UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF FLORIDA

CASE NO. _____

K.MIZRA LLC,

Plaintiff,

v.

CITRIX SYSTEMS, INC., AND
CLOUD SOFTWARE GROUP, INC.,

Defendants.
_____/

COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff K.Mizra LLC ("K.Mizra") files this Complaint for patent infringement against Defendants Citrix Systems, Inc. ("Citrix Systems") and Cloud Software Group Inc. ("Cloud Software Group", and collectively with Citrix Systems "Defendants" or "Citrix"), alleging as follows:

I. INTRODUCTION

1. K.Mizra is a patent licensing company run by experienced management. The company focuses on high value, high quality patents with a global reach. It owns patent portfolios originating with a wide array of inventors, including portfolios developed by well-known multinationals such as IBM, Intel, Rambus and others, as well as from research institutes such as Nederlandse Organisatie voor Toegespast Natuurwetenschappelijk Onderzoek (Netherlands Organization for Applied Scientific Research). By focusing on high quality patents, K.Mizra provides a secondary market for inventors to recoup their research and development investments and to continue their innovations. K.Mizra offers licenses to its patents on reasonable terms and in this way plays an important part in the development of the technologies that improve all our lives.

2. K.Mizra is the owner by assignment of United States Patent No. 8,234,705 ("the '705 Patent" or "the Asserted Patent"). The Asserted Patent was involved in an unsuccessful *Inter Partes* Review Proceeding ("IPR") and several now-resolved federal court litigations, and was originally invented by two highly respected and prolific individual inventors, James A. Roskind and Aaron T. Emigh.

3. The Asserted Patent was originally owned by Dr. Roskind and Mr. Emigh's company, Radix Labs, LLC. Dr. Roskind and Mr. Emigh were then, and remain today, focused on innovation, conducting new research, developing new technologies, and creating new and innovative computer products.

4. Dr. Roskind, one of the two inventors of the Asserted Patent, has bachelor's, master's, and doctorate degrees from the Massachusetts Institute of Technology in both electrical engineering and computer science, and is the named inventor of over 300 U.S. patents. He has worked for Netscape as the Chief Architect and as the Netcenter Security Architect and was a co-founder for Infoseek, a company that was eventually acquired by Disney for $770 million. He was also a key developer of Google's "transport protocol" that provides the tech giant billions of dollars in value every year.

5. Mr. Emigh, the other named inventor of the Asserted Patent, graduated from the University of California, Santa Cruz with degrees in linguistics and computer and information sciences, and is the named inventor of over 140 patents. Prior to working with Dr. Roskind, Mr. Emigh worked in various positions developing software, including working as a software manager, architect, and engineer for Unicom and working as a manager for the software development and technical marketing groups for Philips TriMedia. He has founded or co-founded

K.Mizra LLC, Ex. 2001, p. 2
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

many companies, in addition to Radix Labs, LLC, including CommerceFlow, Inc., which was acquired by eBay for its technology that Mr. Emigh helped develop.

6. After the Asserted Patent issued, Dr. Roskind and Mr. Emigh recouped their research and development investment by selling their rights thereto and continued on in their individual technology development pursuits. K.Mizra ultimately acquired the Asserted Patent and licensed it to many of the who's-who of the tech world. Some of the accused infringers chose to test the validity of the Asserted Patent before settling their lawsuits involving the Asserted Patent. For instance, a few accused infringers of the Asserted Patent previously sought IPR by the Patent Trial and Appeal Board ("PTAB"). A Final Written Decision ("Decision") in the IPR found that the petitioners had not shown, by a preponderance of the evidence, that the asserted claims were unpatentable. The IPR Decision was appealed to the U.S. Court of Appeals for the Federal Circuit ("CAFC"), resulting in a procedurally focused remand to the PTAB. Prior to the issuance of the mandate that would have sent the IPR back to the United States Patent and Trademark Office ("USPTO") for further consideration, the parties agreed to move to dismiss the appeal.

7. K.Mizra remains ready, willing, and able to provide commercially-reasonable licenses for its various patented technologies to all entities who wish or need to use them internally or in connection with products or services offered to others. As outlined below, Citrix is one such entity.

II. THE PARTIES

8. K.Mizra is a Delaware limited liability company with a mailing address of 777 Brickell Avenue, #500-96031, Miami, Florida 33131, and operates in Florida. K.Mizra is the owner by assignment of the Asserted Patent.

K.Mizra LLC, Ex. 2001, p. 3
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

9. Citrix Systems is a corporation organized and existing under Delaware law with a principal place of business at 851 W. Cypress Creek Road, Fort Lauderdale, Florida 33309. *See* https://www.citrix.com/contact/offices.html (last accessed April 23, 2025), a true and correct copy of which is attached as Exhibit 1. This exhibit, and all other exhibits referenced in this Complaint, are incorporated by reference in their entireties. On information and belief, Citrix Systems is a subsidiary and/or business unit of Cloud Software Group.

10. Citrix Systems may be served through its registered agent, Corporation Service Company, 1201 Hays Street, Tallahassee, Florida 32301-2525.

11. Cloud Software Group is a corporation organized and existing under Delaware law with a principal place of business at 851 W. Cypress Creek Road, Fort Lauderdale, Florida 33309. A printout from the website of the Florida Department of State, Division of Corporations, showing details for Cloud Software Group, Inc., is attached as Exhibit 2.

12. On information and belief, Cloud Service Group operates a sales office for Citrix products, including products including the Secured Private Access solution, at 851 West Cypress Creek Road, Fort Lauderdale, Florida 33309.

13. Cloud Software Group may be served through its registered agent, CT Corporation System, 1200 S. Pine Island Road, Plantation, Florida 33324.

III. JURISDICTION AND VENUE

14. This is an action for patent infringement under the patent laws of the United States, 35 U.S.C. §§ 1 *et seq.*, including 35 U.S.C. §§ 271, 281, and 284, among others. The Court has subject-matter jurisdiction over the claims raised in this action pursuant to 28 U.S.C. §§ 1331 and 1338(a).

K.Mizra LLC, Ex. 2001, p. 4
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

15. This Court has personal jurisdiction over Citrix by virtue of, *inter alia*, its principal place of business in Fort Lauderdale, Florida; its appointment of a registered agent in Florida; its conduct of business in this District; its purposeful availment of the rights and benefits of Florida law; and its substantial, continuous, and systematic contacts with the state of Florida and this District. Citrix further: (1) intentionally markets and sells its infringing products directly and through agents to residents of Florida; (2) enjoys substantial income from the state of Florida; and/or (3) directly, by its own actions, and/or in combination with actions of customers and others under its control, has committed acts of infringement in this District at least by making and using infringing systems and using, selling, and offering for sale infringing services.

16. Venue is proper in this District pursuant to 28 U.S.C. § 1400(b) because Citrix has its principal place of business (which it designates as its corporate headquarters) in the state of Florida and in this District.

IV. GENERAL ALLEGATIONS

A. The Asserted Patent

17. K.Mizra is the sole owner by assignment of the Asserted Patent with the full and exclusive right to bring suit to enforce them. (*See* Ex. 3.) K.Mizra is also entitled to sue to collect damages for all past infringement of the Asserted Patent.

18. The '705 Patent, titled "Contagion Isolation and Inoculation," was legally issued by the USPTO to inventors Dr. Roskind and Mr. Emigh on July 31, 2012. A true and correct copy of the '705 Patent is attached hereto as Exhibit 4.

19. The Asserted Patent claims priority to U.S. Provisional Application No. 60/613,909, filed on September 27, 2004 (the "Provisional Application").

K.Mizra LLC, Ex. 2001, p. 5
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

B. Prior Licensing And Litigation Of The Asserted Patent

20. The Asserted Patent has been owned by several entities, in addition to Radix Labs, LLC and K.Mizra, with some of those entities issuing to third parties certain rights to the technologies covered thereby.

21. K.Mizra has been involved in a number of actions it had to institute to protect its patent rights, including actions involving the Asserted Patent. Most of those actions resulted in the execution of confidential patent license agreements.

22. Citrix is not and has never been a licensee of the Asserted Patent nor had or has any rights to use technologies covered by the Asserted Patent. Citrix thus has no ownership or other rights (and is entitled to no rights) relating to the Asserted Patent.

C. Computer Network Security Problems In 2004 Solved By The Asserted Patent

23. The technology described in the Asserted Patent was invented by Dr. Roskind and Mr. Emigh, two colleagues living in the same area who had similar interests in innovating computer-related technologies. In 2003, the inventors decided to create a business—Radix Labs, LLC—which focused on developing intellectual property related to various computer technologies, including computer network security technologies. The inventors focused on conceiving and reducing to practice inventions that they knew were needed (or soon would be needed) in the computer networking industry and then on drafting patent applications to capture and protect their technological innovations. In September 2004, the inventors filed the Provisional Application to which the Asserted Patent claims priority. The Provisional Application described technology that focused on securing a computer network against the threats to which it was exposed when computer endpoints (e.g., laptop computers) were connected to a computer network. The Provisional Application, and by natural extension the Asserted Patent, also focuses on

K.Mizra LLC, Ex. 2001, p. 6
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

remedying identified threats and quarantining those threats to mitigate any damage to the secured network.

24. Claims of the Asserted Patent are directed to technological solutions that address specific challenges grounded in computer network security. Maintaining the security of computer systems and networks is a tremendous concern for modern enterprises, since a breach of an internal network can have severe repercussions, including major financial losses, data theft, disclosure of sensitive information, network disruptions, data corruption, etc. The inventors of the Asserted Patent understood that while a network security appliance or hardware can be adept at keeping out unwanted external intrusions from the network, the most exploitable vulnerabilities of most networks are the end-user computers that roam throughout various public and private network domains, potentially exposing those computers to infection and then accessing and potentially infecting the entire and presumably secure computer network.

25. For example, the '705 Patent explains that "[l]aptop and wireless computers and other mobile systems pose a threat to elements comprising and/or connected to a network service provider, enterprise, or other protected network to which they reconnect after a period of connection to one or more networks and/or systems that are not part of the service provider, enterprise, or other protected network. By roaming to unknown domains, such as the Internet, and/or connecting to such domains through public, wireless, and/or otherwise less secure access nodes, such mobile systems may become infected by computer viruses, worms, backdoors, and/or countless other threats and/or exploits and/or have unauthorized software installed; have software installed on the mobile system by an operator of the protected network for the protection of the mobile system and/or the protected network removed or altered without authorization and/or have

K.Mizra LLC, Ex. 2001, p. 7
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

configurations, settings, security data, and/or other data added, removed, and/or changed in authorized ways and/or by unauthorized person[s]." (*See, e.g.*, Ex. 4 at 1:14–31.)

26. The solution to these problems—as specified and claimed in the Asserted Patent—was an advanced departure from the conventional network access control solutions then in use and was then, as it remains today, patent eligible, highly valuable, novel, and non-obvious technology.

D. K.Mizra's Asserted Patent Claims Are Presumed Valid

27. K.Mizra asserts that at least, and without limitation, Claim 19 of the '705 Patent has been directly infringed, either literally or under the doctrine of equivalents. K.Mizra reserves the right to assert additional claims of the Asserted Patent, including both independent and dependent claims, pursuant to the Court's (and other applicable) rules and procedures and as discovery progresses. These claims are referred to herein as the "Asserted Claims."

28. None of the Asserted Claims are directed to abstract ideas, and each employs inventive concepts and is directed to patent-eligible subject matter. All claims of the Asserted Patent are also presumed to be valid and enforceable against Citrix and others.

29. Indeed, the Asserted Patent's specification and claims demonstrate that the need satisfied by the inventions of the Asserted Claims was long-felt in the industry and thus unconventional. As one example, the '705 Patent explains that mobile end user devices such as laptops and wireless computers pose a threat to protected network elements because those devices may access unsecure systems and thereby "become infected by computer viruses, worms, backdoors, and/or countless other threats and/or exploits and/or have unauthorized software installed; have software installed on the mobile system by an operator of the protected network for the protection of the mobile system and/or the protected network removed or altered without authorization; and/or have configurations, settings, security data, and/or other data added,

K.Mizra LLC, Ex. 2001, p. 8
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

removed, and/or changed in unauthorized ways and/or by unauthorized person[s]." (Ex. 4 at 1:23-31.) Similarly, stationary systems such as desktop computers "may become infected, e.g., due to receipt and execution of malicious code via a network or other communication and/or a diskette and/or other removable media." (*Id.* at 1:31-34.) This poses a danger to a protected network because, when the user device connects to the protected network, "a system may infect or otherwise harm resources associated with the protected network before measures can be taken to detect and prevent the spread of such infections or harm." (*Id.* at 1:34-38.) The Asserted Patent's specification further provides that "[t]herefore, there is a need for a reliable way to ensure that a system does not infect or otherwise harm other network resources when connected to a protected network." (*Id.* at 1:38-41.)

30. The specification (including the provisions quoted above), the figures (including those included below), and the text related to the figures further illustrate the complex, tiered network system architecture of the inventions captured by the Asserted Claims. These figures include the following:

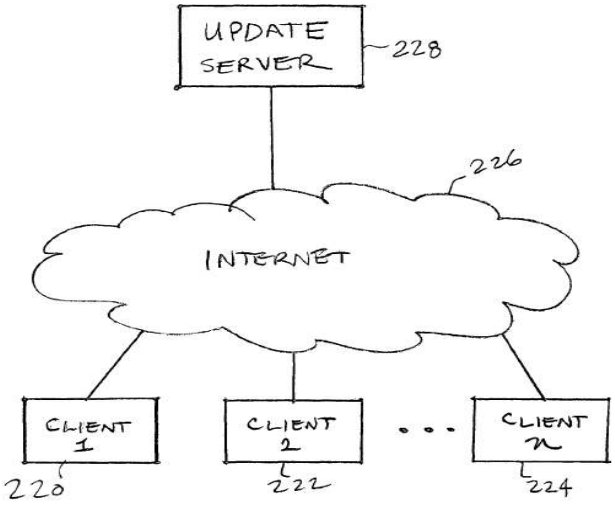
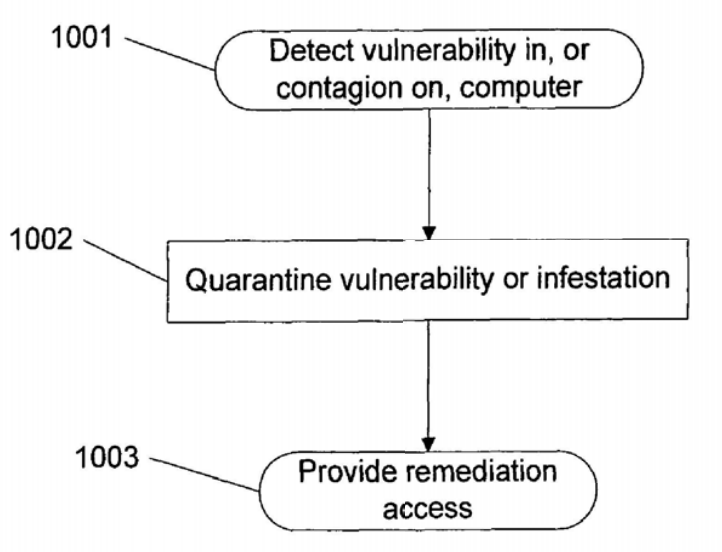


FIG. 2B

(*See* Ex. 4 at Fig. 2B.)

K.Mizra LLC, Ex. 2001, p. 9
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468



(*See id.* at Fig. 10A.)

31. The foregoing demonstrates that the inventions of the Asserted Claims focus on specific tamperproof hardware that must interact with unique software to improve network access control technology and protect a secure computer network and the data stored thereon from infected devices. Thus, the Asserted Claims are eligible as a matter of law for patent protection under step one of *Alice Corp. v. CLS Bank Int'l*, 573 U.S. 208, 216 (2014).

32. All actions and steps recited in the Asserted Claims, including the act of quarantining endpoints or other computers, if necessary, requires the involvement of various hardware components running dedicated software both before, during, and after the selection and isolation of an object. Said another way, a claim directed to allowing a machine to automatically and dynamically select and isolate an unsafe device attempting to access a secure network is not simply adding a generic computer component to a fundamentally human process. Rather, it is removing the once-necessary human intervention from a fundamentally mechanical process, an "improvement in the functioning of a" networked system that simply cannot be considered directed to an abstract concept. *Enfish LLC v. Microsoft Corp.*, 822 F.3d 1327, 1339 (Fed. Cir. 2016).

K.Mizra LLC, Ex. 2001, p. 10
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

33. As the specification confirms, the improvement captured by the Asserted Claims is not simply quarantining an infected device, but it is instead a multi-faceted network system involving multiple interrelated software and hardware components to protect a network from known and unknown threats. Specifically, the specification of the Asserted Patent discloses that to reduce the burdens of having to manually identify, connect to, isolate, and remove malicious software from an infected device, the networked system can direct an unclean computer attempting to connect to the secure network, known as the host computer, to a form of remediation, such as downloading a software patch or a software update, removing material from the host computer and/or enabling certain settings, etc. present on the host computer. (*See* Ex. 4 at 1:14–41.) Indeed, the inventions of the Asserted Claims are each tethered to these advances over the art in the 2005 time frame, reciting methods and systems that automatically and dynamically detect an insecure condition by contacting a trusted computing base, receiving a response therefrom, determining whether that response contains a valid identification of cleanliness, and configuring and implementing a remediation action based on what is discovered about the state of an endpoint or "host" computer. (*See, e.g.*, Ex. 4, Claims 12 and 19.) More specifically, the Asserted Claims require a system configured to communicate with a "trusted computing base" to determine when a response includes a valid digitally signed attestation of cleanliness, and to control access to the network accordingly. These Asserted Claims are thus directed to a machine-implemented solution resolving a machine-specific problem: a machine's difficulty in detecting, isolating, and remediating infected endpoint devices (*e.g.*, host computers) to prevent contagion of and damage to the larger computer network.

34. The Asserted Claims are thus directed to a machine-implemented process for (1) determining whether the host computer is required to be quarantined, (2) isolating and inoculating

K.Mizra LLC, Ex. 2001, p. 11
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

the contagions (including directing the host to software programs and/or code designed to identify undesirable and/or unauthorized states) by quarantining the host, (3) limiting access to the network by the host computer so that the unsafe condition thereof can be remedied, and (4) allowing for remediation of an unsafe or infected host computer. As such, the Asserted Claims recite inventions with specific applications or improvements to technologies in the marketplace and cannot be considered abstract or patent ineligible under relevant law.

E. Failed IPR

35. Fortune 100 companies accused of infringing the Asserted Patent have previously filed petitions for IPRs, alleging that the claims of the Asserted Patent should be held invalid as either anticipated or obvious considering art not previously considered. Ultimately, the PTAB instituted an IPR against the '705 Patent, with similar third party IPRs that were subsequently filed being joined to the first-filed and instituted IPR.

36. The PTAB eventually issued its decision holding that no claims of the '705 Patent were unpatentable, finding that no asserted prior art reference alone or in combination satisfied the limitation of "providing . . . an IP address of a quarantine server configured to serve the quarantine notification page" that was present in all claims of the '705 Patent.

37. The IPR Decision was then appealed to the CAFC, which reversed the PTAB's Decision on a few narrow procedural issues involving proof that the asserted prior art references would be combined by a person having ordinary skill in the art, as alleged by the petitioners.

38. The IPR involving the '705 Patent has since been dismissed by the PTAB at the request of the parties.

K.Mizra LLC, Ex. 2001, p. 12
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

F. Citrix's Accused Instrumentalities And Services

39. Citrix has been making, selling, using, and offering for sale computer network security products and services that infringe the Asserted Patent in violation of 35 U.S.C. § 271. These include, but are not limited to, Citrix's Secure Private Access (SPA) solution (including native cloud and/or on premises implementations) and/or Citrix products including the SPA solution (the "Accused Instrumentalities"), the sale, offer for sale, use and/or manufacture in the United States of which constitutes infringement of at least and without limitation, the Asserted Claims directly, either literally or under the doctrine of equivalents (DoE).

G. K.Mizra's Efforts To Work With Citrix

40. K.Mizra has contacted Citrix on several occasions seeking to discuss Citrix's infringement. For example, K.Mizra contacted Citrix at least as early as August 2022 in a letter addressing Citrix's potential infringement and directing Citrix to the list of K.Mizra-owned patents found on its website. K.Mizra followed up on this letter in April 2023.

41. K.Mizra subsequently sent a letter to Citrix which was delivered on January 31, 2025. A copy of the letter is attached as Exhibit 5. The letter was accompanied by a claim chart showing how the Citrix SPA solution infringes the Asserted Patent. Citrix did not respond to this letter.

COUNT I
(Patent Infringement under 35 U.S.C. § 271 of the '705 Patent)

42. K.Mizra incorporates paragraphs 1 through 41 as though fully set forth herein.

43. The '705 Patent includes 19 claims.

44. Citrix has directly infringed one or more claims of the '705 Patent by making, importing, using, offering for sale, and/or selling the Accused Instrumentalities, all in violation of 35 U.S.C. § 271(a).

K.Mizra LLC, Ex. 2001, p. 13
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

45. Based on publicly available information, the Accused Instrumentalities satisfy every element of at least Claim 19 of the '705 Patent.

46. For example, Claim 19 of the '705 Patent states:

> [preamble] A computer program product for protecting a network, the computer program product being embodied in a non-transitory computer readable medium and comprising computer instructions for:
>
> [A] detecting an insecure condition on a first host that has connected or is attempting to connect to a protected network,
>
> [B] wherein detecting the insecure condition includes:
>
> [B1] contacting a trusted computing base associated with a trusted platform module within the first host,
>
> [B2] receiving a response, and determining whether the response includes a valid digitally signed attestation of cleanliness,
>
> [C] wherein the valid digitally signed attestation of cleanliness includes at least one of an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation that the trusted computing base has ascertained the presence of a patch or a patch level associated with a software component on the first host;
>
> [D] when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantining the first host, including by preventing the first host from sending data to one or more other hosts associated with the protected network,
>
> [E] wherein preventing the first host from sending data to one or more other hosts associated with the protected network includes
>
> [E1] receiving a service request sent by the first host, serving a quarantine notification page to the first host when the service request comprises a web server request,
>
> [E2] and in the event the service request comprises a DNS query, providing in response an IP address of a quarantine server configured to serve the quarantine notification page if a host name that is the subject of the DNS query is not

K.Mizra LLC, Ex. 2001, p. 14
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

> associated with a remediation host configured to provide data usable to remedy the insecure condition; and
>
> [F] permitting the first host to communicate with the remediation host.

(Ex. 4 at 22:14-49.)

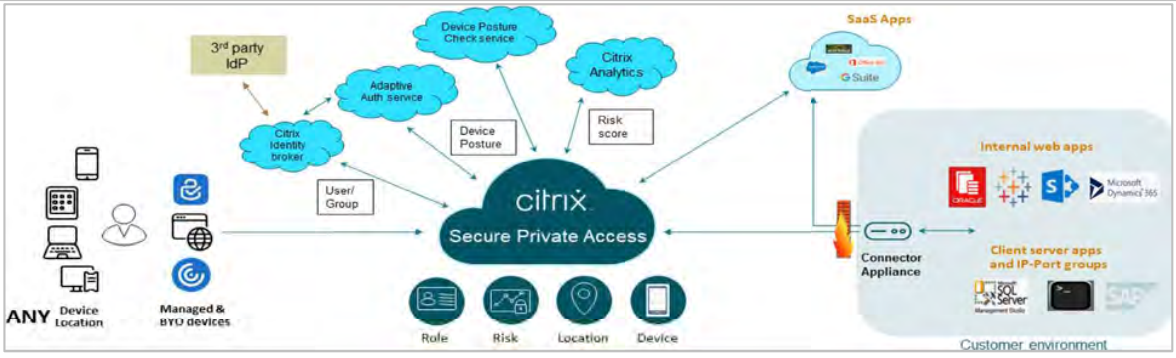
47. As for the preamble of Claim 19, to the extent that it is determined to be limiting, the Accused Instrumentalities provide the features described in the preamble, which recites a "computer program product for protecting a network." Citrix's SPA solution provides for secure access to protected network resources whether stored in the cloud or in an on-premises datacenter.

Citrix solutions for ZTNA

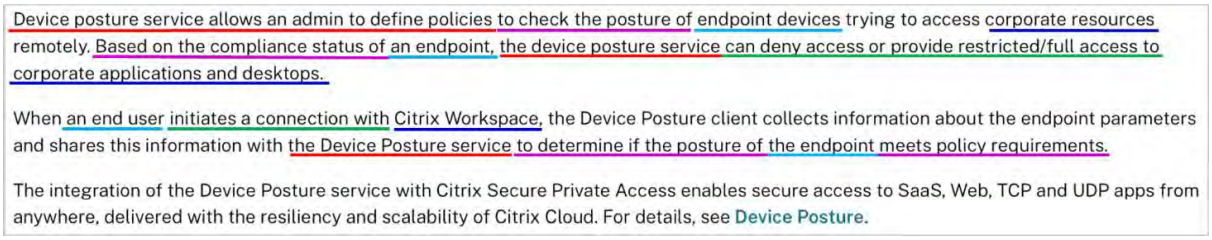
With Citrix, you can deliver secure access to managed, unmanaged, and BYOD devices alike — without compromising the end user experience. Citrix Secure Private Access provides adaptive access to all corporate applications, whether they're deployed in the cloud or an on-premises datacenter. This cloud-based ZTNA solution provides access only at the application level, allowing you to strengthen your security posture and replace your VPN to avoid common issues like network-level attacks.

(*See* What is zero trust network access (ZTNA)?, available at https://www.citrix.com/glossary/what-is-zero-trust-network-access.html (last accessed April 23, 2025).) For example, Citrix touts that its SPA product "offers Adaptive Authentication, single sign-on support, enhanced security controls for the applications."

Citrix Secure Private Access offers Adaptive Authentication, single sign-on support, enhanced security controls for the applications. Secure Private Access also provides the capabilities to scan the end user device before establishing a session by using the Device Posture service. Based on the Adaptive Authentication or Device Posture results, admins can define the authentication methods for the apps.



(*See* Ex. 6, Secure Private Access (SPA) (available at https://docs.citrix.com/en-us/citrix-secure-private-access/service/spa-solution-overview.html) (published December 5, 2024) (last accessed April 23, 2025).) The SPA solution includes a Device Posture service that allows the SPA solution to ensure that the end user device meets certain criteria before connecting to the protected network.

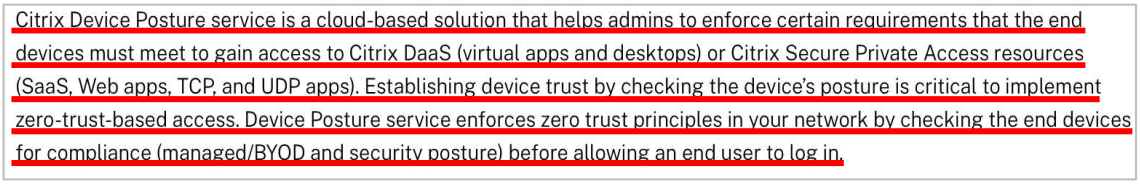


Device posture service allows an admin to define policies to check the posture of endpoint devices trying to access corporate resources remotely. Based on the compliance status of an endpoint, the device posture service can deny access or provide restricted/full access to corporate applications and desktops.

When an end user initiates a connection with Citrix Workspace, the Device Posture client collects information about the endpoint parameters and shares this information with the Device Posture service to determine if the posture of the endpoint meets policy requirements.

The integration of the Device Posture service with Citrix Secure Private Access enables secure access to SaaS, Web, TCP and UDP apps from anywhere, delivered with the resiliency and scalability of Citrix Cloud. For details, see Device Posture.

(*Id.*) Citrix explains that the "Device Posture service is a cloud-based solution that helps admins to enforce certain requirements that the end devices must meet to gain access to Citrix DaaS (virtual apps and desktops) or Citrix Secure Private Access resources (SaaS, Web apps, TCP, and UDP apps)."



Citrix Device Posture service is a cloud-based solution that helps admins to enforce certain requirements that the end devices must meet to gain access to Citrix DaaS (virtual apps and desktops) or Citrix Secure Private Access resources (SaaS, Web apps, TCP, and UDP apps). Establishing device trust by checking the device's posture is critical to implement zero-trust-based access. Device Posture service enforces zero trust principles in your network by checking the end devices for compliance (managed/BYOD and security posture) before allowing an end user to log in.

(*See* Ex. 7, Device Posture Overview (available at https://docs.citrix.com/en-us/device-posture/device-posture-overview) (published March 3, 2025) (last accessed April 23, 2025).) Citrix's Secure Private Access solution is available as a cloud native service, or can be downloaded and managed by the customer. (*See* https://docs.citrix.com/en-us/citrix-secure-private-access (last accessed April 23, 2025); *see also* Deployment Guide: Citrix Secure Private Access On-Premises, available at https://community.citrix.com/tech-zone/build/deployment-guides/secure-private-access-on-premises/ (last accessed April 23, 2025).) The Secure Private Access solution can be downloaded through the Citrix website at https://www.citrix.com/downloads/citrix-secure-

K.Mizra LLC, Ex. 2001, p. 16
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

private-access/ (last accessed April 23, 2025). Accordingly, and to the extent that the preamble of Claim 19 is somehow limiting, the Accused Instrumentalities would meet the limitation.

48. Limitation A of Claim 19 requires "detecting an insecure condition on a first host that has connected or is attempting to connect to a protected network." The Accused Instrumentalities also meet all the requirements of limitation A of Claim 19. Citrix's SPA solution "gives IT a set of security controls to protect against threats from BYO devices, giving the users the choice to access their IT-sanctioned applications from any device, whether its managed or BYO." (Ex. 6 at p. 1.) Among these security controls is that the SPA solution "provides the capabilities to scan the end user device before establishing a session by using the Device Posture Service." For example, Citrix's SPA product "provides the capability to scan the end user device before establishing a session by using the Device Posture service."

> Citrix Secure Private Access offers Adaptive Authentication, single sign-on support, enhanced security controls for the applications. Secure Private Access also provides the capabilities to scan the end user device before establishing a session by using the Device Posture service. Based on the Adaptive Authentication or Device Posture results, admins can define the authentication methods for the apps.

(Ex. 6 at p. 1.) For example, when the end user initiates a connection with Citrix's workspace, information is collected about the endpoint user's device parameters to detect an insecure condition of the endpoint device.

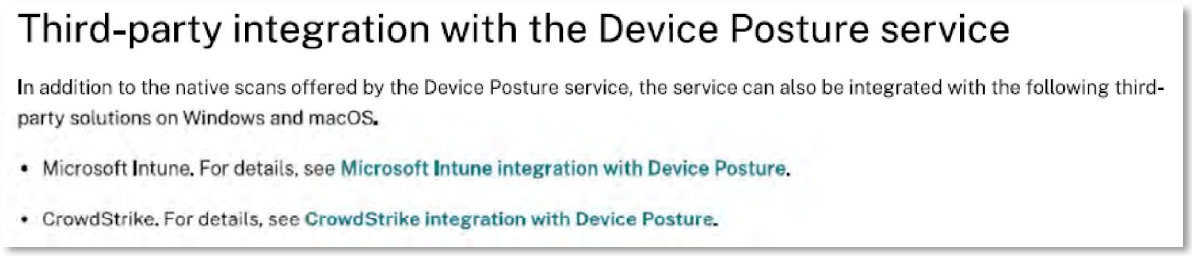
> Device posture service allows an admin to define policies to check the posture of endpoint devices trying to access corporate resources remotely. Based on the compliance status of an endpoint, the device posture service can deny access or provide restricted/full access to corporate applications and desktops.
>
> When an end user initiates a connection with Citrix Workspace, the Device Posture client collects information about the endpoint parameters and shares this information with the Device Posture service to determine if the posture of the endpoint meets policy requirements.
>
> The integration of the Device Posture service with Citrix Secure Private Access enables secure access to SaaS, Web, TCP and UDP apps from anywhere, delivered with the resiliency and scalability of Citrix Cloud. For details, see Device Posture.

(*See id.* at p. 2.)

49. Limitation B1 of Claim 19 requires that "detecting [an] insecure condition includes . . . contacting a trusted computing base associated with a trusted platform module within the first host." The Accused Instrumentalities meet these requirements by using the Device Posture

K.Mizra LLC, Ex. 2001, p. 17
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

service of the Secure Protected Access solution. For example, the Citrix SPA solution supports integration of the Device Posture service with Microsoft Intune.

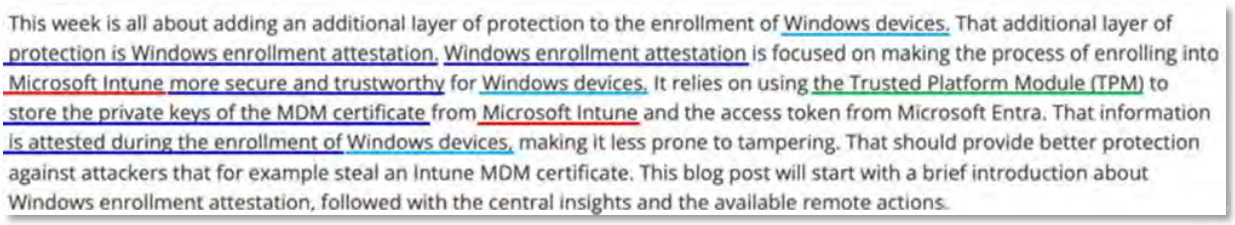


Third-party integration with the Device Posture service

In addition to the native scans offered by the Device Posture service, the service can also be integrated with the following third-party solutions on Windows and macOS.

- Microsoft Intune. For details, see Microsoft Intune integration with Device Posture.
- CrowdStrike. For details, see CrowdStrike integration with Device Posture.

(Ex. 7 at p. 4.) When integrating with Microsoft Intune, the Citrix SPA solution is configured to contact a trusted computing base associated with a trusted platform module within the end user device, ensuring that only secure, compliant devices can access network resources.



This week is all about adding an additional layer of protection to the enrollment of Windows devices. That additional layer of protection is Windows enrollment attestation. Windows enrollment attestation is focused on making the process of enrolling into Microsoft Intune more secure and trustworthy for Windows devices. It relies on using the Trusted Platform Module (TPM) to store the private keys of the MDM certificate from Microsoft Intune and the access token from Microsoft Entra. That information is attested during the enrollment of Windows devices, making it less prone to tampering. That should provide better protection against attackers that for example steal an Intune MDM certificate. This blog post will start with a brief introduction about Windows enrollment attestation, followed with the central insights and the available remote actions.

(*See* Ex. 8, Getting started with Windows enrollment attestation, p. 1 (available at https://petervanderwoude.nl/post/getting-started-with-windows-enrollment-attestation/) (published September 30, 2024) (last accessed April 23, 2025).) The TPM provides a secure hardware environment to protect the certificates, safeguarding them from tampering and ensuring that only TPM-validated, certificate-verified devices can connect to sensitive applications and data.

K.Mizra LLC, Ex. 2001, p. 18
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

> The goal of Windows enrollment attestation is to make devices more secure and trustworthy within the network they join. With this feature, you can check that Windows 10 and 11 devices meet strict security standards during enrollment, using Trusted Platform Module (TPM) technology to enhance their defense against threats. The Windows enrollment attestation feature also confirms and reports on the devices that enroll securely, ensuring the process is reliable.
>
> Here's how it benefits organizations:
>
> **Improved security**: TPM attestation helps detect and address security weaknesses or compromised devices and lowers the chance of unauthorized access or security incidents.

(*See* Windows enrollment attestation, available at https://learn.microsoft.com/en-us/intune/intune-service/enrollment/windows-enrollment-attestation (published March 3, 2025) (last accessed April 23, 2025).) As another example, the Device Posture service may contact a Citrix Device Posture client, also known as the EPA client, that is installed on an end user device.

- Citrix Device Posture client (EPA client): A lightweight application that must be installed on the endpoint device to run device posture scans. This application does not require local admin rights to download and install on an endpoint.

(Ex. 7 at p. 2.) Although public information does not confirm whether or not the EPA client uses Trusted Platform Module technology, use of such technology is a way to accomplish the ends of the EPA client in a secure manner. Discovery is needed to confirm the detailed operation of the EPA client in connection with the SPA solution.

50. Limitation B2 of Claim 19 requires that "detecting the insecure condition" also includes "receiving a response and determining whether the response includes a valid digitally signed attestation of cleanliness." The Accused Instrumentalities also meet all the requirements of limitation B2. For example, when a remote device ("first host") initiates a request to access corporate resources (*i.e.*, a protected network), the Citrix Secure Private Access solution (in conjunction with the Device Posture service) determines whether the end user device is complaint with policy requirements.

K.Mizra LLC, Ex. 2001, p. 19
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

Citrix Secure Private Access offers Adaptive Authentication, single sign-on support, enhanced security controls for the applications. Secure Private Access also provides the capabilities to scan the end user device before establishing a session by using the Device Posture service. Based on the Adaptive Authentication or Device Posture results, admins can define the authentication methods for the apps.

(Ex. 6 at p. 1.)

Device posture service allows an admin to define policies to check the posture of endpoint devices trying to access corporate resources remotely. Based on the compliance status of an endpoint, the device posture service can deny access or provide restricted/full access to corporate applications and desktops.

When an end user initiates a connection with Citrix Workspace, the Device Posture client collects information about the endpoint parameters and shares this information with the Device Posture service to determine if the posture of the endpoint meets policy requirements.

The integration of the Device Posture service with Citrix Secure Private Access enables secure access to SaaS, Web, TCP and UDP apps from anywhere, delivered with the resiliency and scalability of Citrix Cloud. For details, see Device Posture.

(*Id.* at p. 2.) To accomplish this, the SPA solution determines whether the response includes a valid digitally-signed attestation of cleanliness.

How Device Attestation Reports Work on the Device

Performing Windows Enrollment Attestation and generating the corresponding report in Intune involves several steps that are seamlessly integrated into the device management workflow. Here's a detailed look at how this process works when users enroll devices into Intune:

1. Device Enrollment and Initialization

When a device is enrolled in device management (Intune), the enrollment process begins with the device requesting a security token from the Intune service. This token is essential for authenticating the device using multifactor authentication and initiating the attestation process.

2. Storing Keys in TPM

Once the device receives the security token, it retrieves the Intune Device Certificate. The critical step here is to store the certificate's enrollment keys, including the private key, in the TPM chip. This step, referred to as **UseTPMForEnrollmentKey**, ensures that the keys are securely stored, protecting them from potential tampering.

3. Initiating Attestation via Microsoft Graph

Intune uses the Microsoft Graph API to initiate the **TPM attestation** from the MDM server. The specific API call, **InitiateMobileDeviceManagementKeyRecovery**, triggers the MDM Key Recovery and TPM attestation processes. This remote command is sent to the device, instructing it to perform the necessary attestation steps.

4. Device Recovery and Attestation

Upon receiving the command, the device executes the recovery process. If the Intune Device Certificate is not already in the TPM, the device recovers it to the TPM. Following this, the device performs the MDMClientCertAttestation with Intune, completing the attestation process.

5. Generating and Viewing Attestation Reports

After the device attestation is performed, the results are compiled into a comprehensive report. IT administrators can view these reports within the Intune portal, providing them with a detailed overview of the attestation status for all managed devices. The reports highlight which devices have successfully completed attestation and which have not, allowing administrators to enforce compliance policies effectively.

(*See* Enhancing Device Security with Windows Enrollment Attestation, available at https://patchmypc.com/windows-enrollment-attestation-tpm-device-attestation (published July 16, 2024) (last accessed April 23, 2025).)

K.Mizra LLC, Ex. 2001, p. 20
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

> **Compliance Reporting** - For Enterprises to reliably match compliance, health, and posture reports with Platforms, they require a durable unique identifier for each Platform. Such an identifier allows Network Administrators to locate, quarantine, or remediate Platforms that have fallen out of compliance with network policy.

(*See* TCG TPM v2.0 Provisioning Guidance, Version 1.0, Revision 1.0, March 15, 2017, available at https://trustedcomputinggroup.org/wp-content/uploads/TCG-TPM-v2.0-Provisioning-Guidance-Published-v1r1.pdf (last accessed April 23, 2025).)

> **2.1 TCG Attestation**
>
> Within the TCG context, Attestation is a process for determining the identity of a device and the software running on the device. Attestation is broken into two phases, shown in Figure 1:
>
> - During system startup, measurements (i.e., hashes computed as fingerprints of files) are "extended", or stored, in the TPM, along with entries added to an informational log. The measurement process generally follows the Chain of Trust model used in Measured Boot, where each stage of the system measures the next one before launching it.
> - Once the device is running and has operational network connectivity, a separate, trusted server (called a Verifier in this document) can interrogate the network device to retrieve the logs and a copy of the digests collected by hashing each software object, signed by a key known only to the TPM.
>
> The result is that the Verifier can verify the device's identity by checking the certificate corresponding to the TPM's attestation key, and can validate the software that was launched by comparing digests in the log with known-good values, and verifying their correctness by comparing with the signed digests from the TPM.
>
> It should be noted that attestation and identity are inextricably linked; signed evidence that a particular version of software was loaded is of little value without cryptographic proof of the identity of the device producing the evidence.[13]

(*See* TCG Remote Integrity Verification: Network Equipment Remote Attestation System, Version 1.0, Revision 9b, June 15, 2019, available at https://trustedcomputinggroup.org/wp-content/uploads/TCG-NetEq-Attestation-Workflow-Outline_v1r9b_pubrev.pdf (last accessed April 23, 2025).)) Thus, the Accused Instrumentalities meet limitation B2 of Claim 19.

51. Limitation C of Claim 19 requires that "the valid digitally signed attestation of cleanliness includes at least one of an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation that the trusted computing base has ascertained the presence of a patch, or a patch level associated with a software component on the first host." The Accused Instrumentalities meet these requirements as the Citrix SPA solution allows administrators to enforce access controls based on device attributes, such as operating system (OS) version or app version.

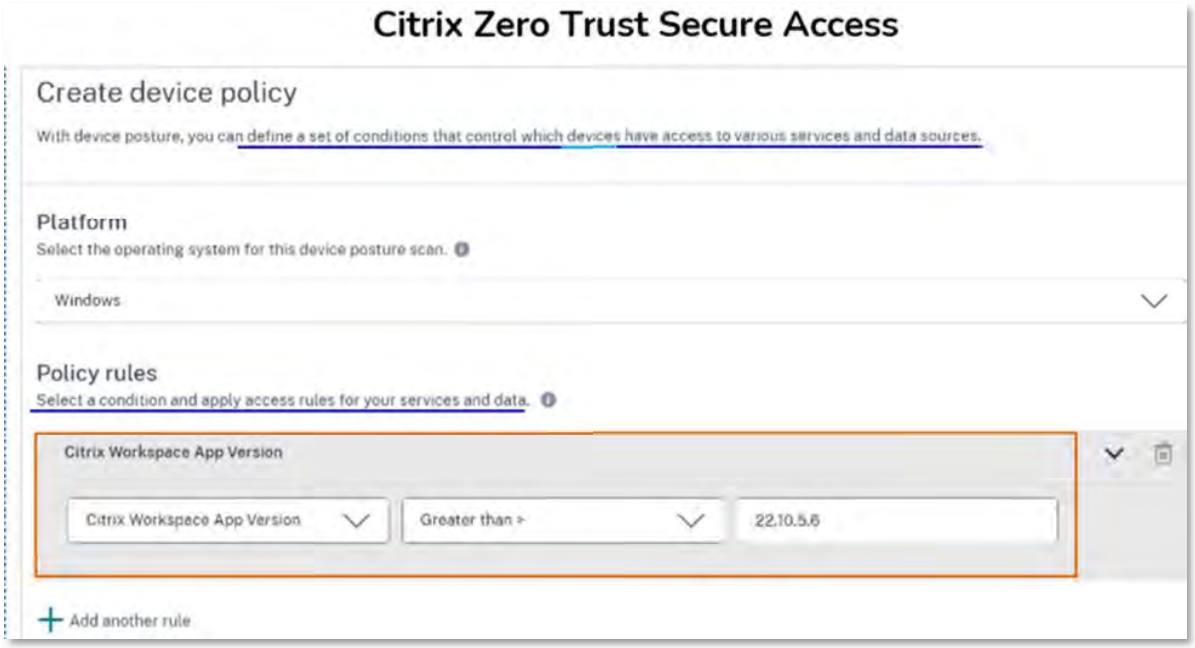
K.Mizra LLC, Ex. 2001, p. 21
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

- The device posture policies must be configured specifically for each platform. For example, for macOS, an admin can allow access for the devices that have a specific OS version. Similarly, for Windows, the admin can configure policies to include a specific authorization file, registry settings, and so on.

(*See* Ex. 7 at p. 3.)

Windows	macOS	iOS	IGEL
Citrix Workspace app version	Citrix Workspace app version	Citrix Workspace app version	-
Operating System version	Operating System version	Operating System version	-
File (exists, file name, and path)	File (exists, file name, and path)	-	File (exists, file name, and path)
Geolocation	Geolocation	-	-
Network location	Network location	-	-
MAC Address	MAC Address	-	-
Process (exists)	Process (exists)	-	-
Microsoft Endpoint Manager	Microsoft Endpoint Manager	-	-
CrowdStrike	CrowdStrike	-	-
Device Certificate	Device Certificate	-	-
Browser	Browser	-	-
Antivirus	Antivirus	-	-
Non-Numeric Registry (32 Bit)	-	-	-
Non-Numeric Registry (64 Bit)	-	-	-
Numeric Registry (32 Bit)	-	-	-
Numeric Registry (64 Bit)	-	-	-
Windows Update Installation Type	-	-	-
Windows Update Installation Last Update Check	-	-	-

(*See id*. at pp. 3-4.)



(*See* Ex. 9, Device Posture, p. 14, (available at https://docs.citrix.com/en-us/device-posture/device-posture.pdf) (published April 22, 2025) (last accessed April 23, 2025).) Citrix's SPA products allow for conditional access based on a set of configurations that control which devices have access to various services and data resources:

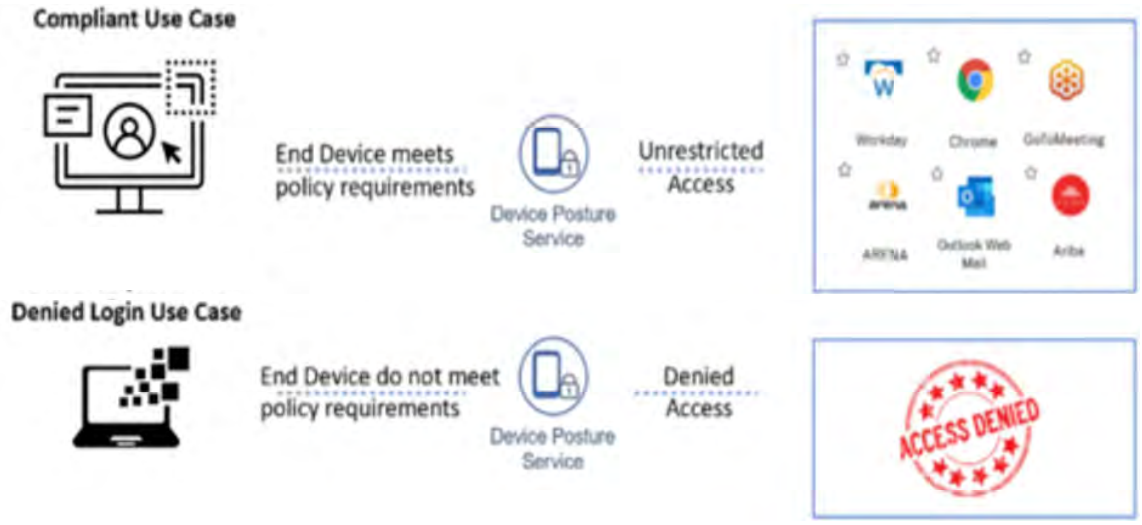
About Device posture

Conditional access is a set of configurations that control which devices have access to various services and data sources. With conditional access, you can create device postures which collect security-related device data, such as OS and browser version, disk encryption and antivirus status. With this data you can define and enforce application access control policies.

(*See* Ex. 10, Device Posture check for Citrix Workspace 2021, pg. 3, (available at https://www.citrix.com/blogs/wp-content/uploads/2023/02/Device-posture-service-handbook.pdf?srsltid=AfmBOoq08-HSo8dPjZ8U6j-6dXC1wSKewGRWTchVPlL-WM-Jtef9yC26) (last accessed April 23, 2025).) Accordingly, the Accused Instrumentalities meet limitation C of Claim 19.

52. Limitation D of Claim 19 requires that "when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantining the first host,

including by preventing the first host from sending data to one or more other hosts associated with the protected network." The Accused Instrumentalities further meet these requirements by having the Citrix SPA products quarantine noncompliant, i.e., unclean, end point devices attempting to connect to the protected network or access corporate resources to prevent potential security threats:



(*See* Ex. 7 at p. 3.)

Device posture service allows an admin to define policies to check the posture of endpoint devices trying to access corporate resources remotely. Based on the compliance status of an endpoint, the device posture service can deny access or provide restricted/full access to corporate applications and desktops.

When an end user initiates a connection with Citrix Workspace, the Device Posture client collects information about the endpoint parameters and shares this information with the Device Posture service to determine if the posture of the endpoint meets policy requirements.

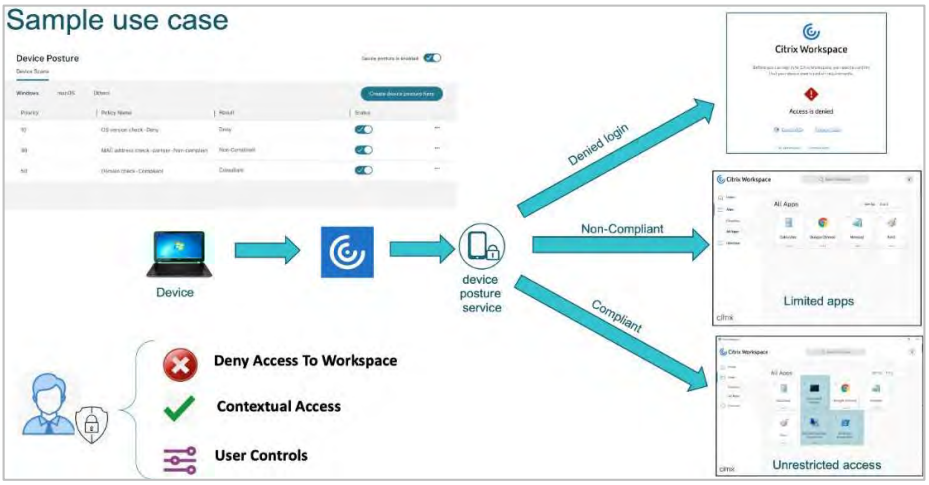
The integration of the Device Posture service with Citrix Secure Private Access enables secure access to SaaS, Web, TCP and UDP apps from anywhere, delivered with the resiliency and scalability of Citrix Cloud. For details, see Device Posture.

(Ex. 6 at p. 2.) Accordingly, the Accused Instrumentalities meet limitation D of Claim 19.

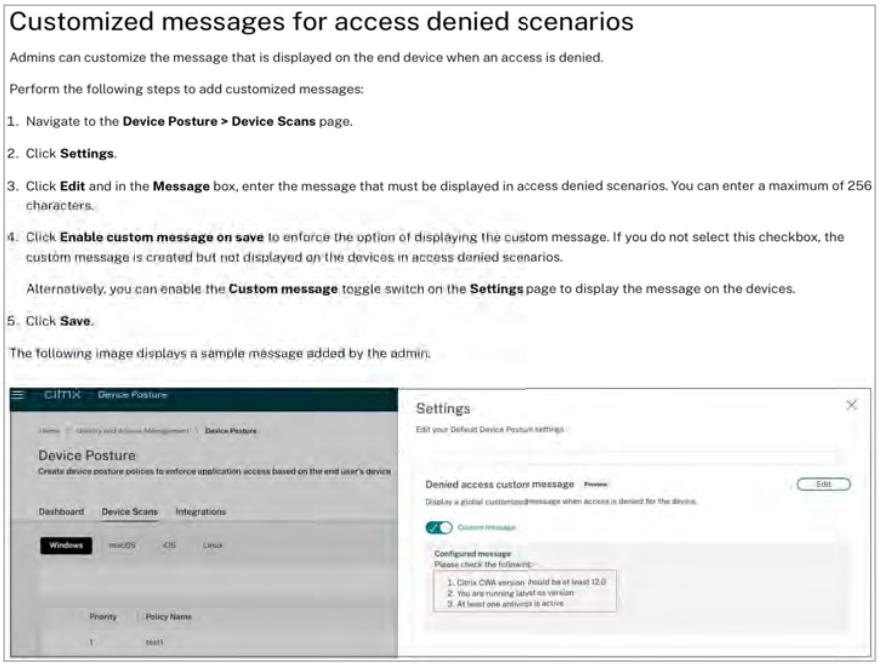
53. Limitation E1 of Claim 19 requires that "preventing the first host from sending data to one or more other hosts associated with the protected network includes . . . receiving a service request sent by the first host [and] serving a quarantine notification page to the first host when the service request comprises a web server request." The Accused Instrumentalities meet these requirements because when Citrix's SPA product employs the Conditional Access policy engine

K.Mizra LLC, Ex. 2001, p. 24
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

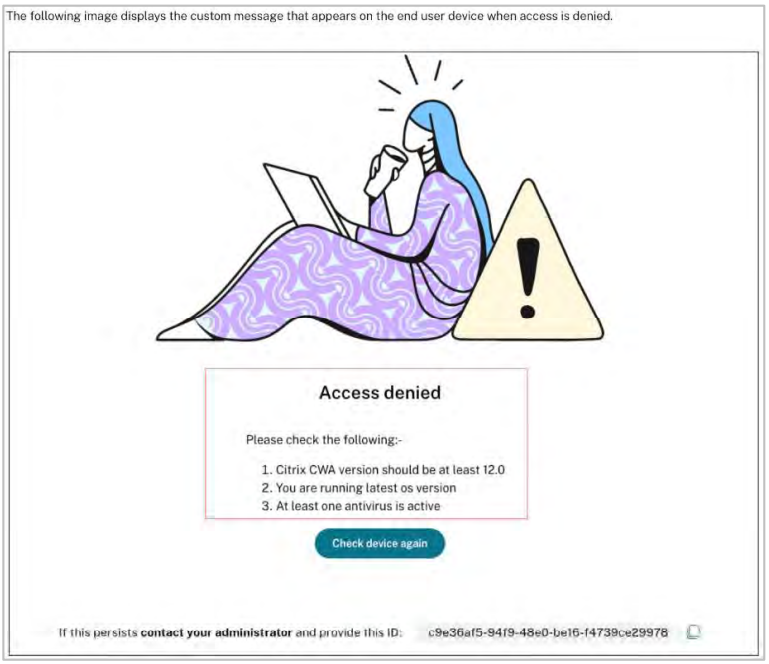
and it determines that a device ("first host") is non-compliant with security policies, the device is restricted from authenticating and accessing company resources and is provided with a notification page ("quarantine notification page").



(*See* Ex. 10 at p. 5). The quarantine notification page may be customized by the customer.



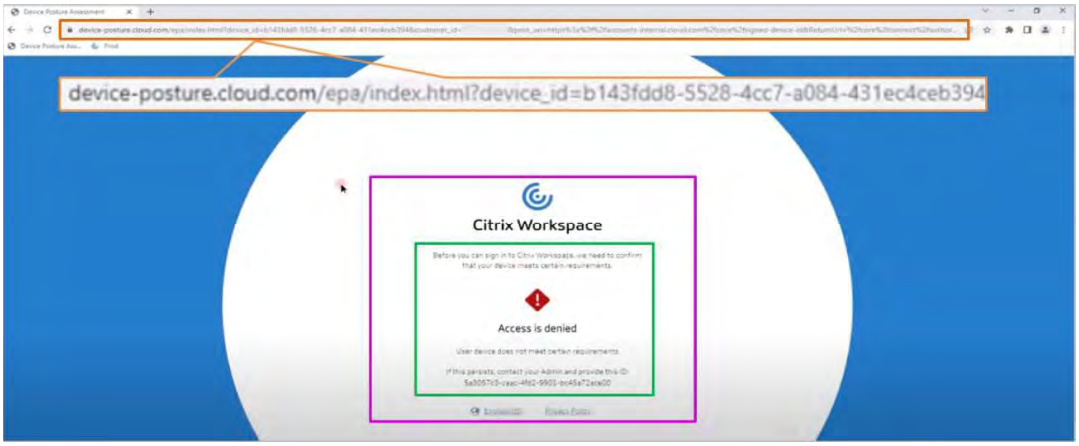
(*See* Ex. 11, Configure Device Posture global settings, p. 4 (published February 21, 2025) (available at https://docs.citrix.com/en-us/device-posture/device-posture-global-settings.html) (last accessed April 23, 2025).)



(*See id.*) Accordingly, the Accused Instrumentalities meet limitation E1 of Claim 19.

54. Limitation E2 of Claim 19 requires that "preventing the first host from sending data to one or more other hosts associated with the protected network includes" "in the event the service request comprises a DNS query, providing in response an IP address of a quarantine server configured to serve the quarantine notification page if a host name that is the subject of the DNS query is not associated with a remediation host configured to provide data usable to remedy the insecure condition." The Accused Instrumentalities also meet all the requirements of limitation E2 of Claim 19.

K.Mizra LLC, Ex. 2001, p. 26
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468



(Screenshot from YouTube video "Integrate Citrix Device Posture Service with Microsoft Intune," available at https://www.youtube.com/watch?v=N85XOxzBtTU (at 1:22) (last accessed April 23, 2025).)

Accordingly, the Accused Instrumentalities meet limitation E2 of Claim 19.

55. Limitation F of Claim 19 requires "permitting the first host to communicate with the remediation host." For example, the SPA solution provides for continuous monitoring which allows the SPA solution to dynamically grant or deny access.

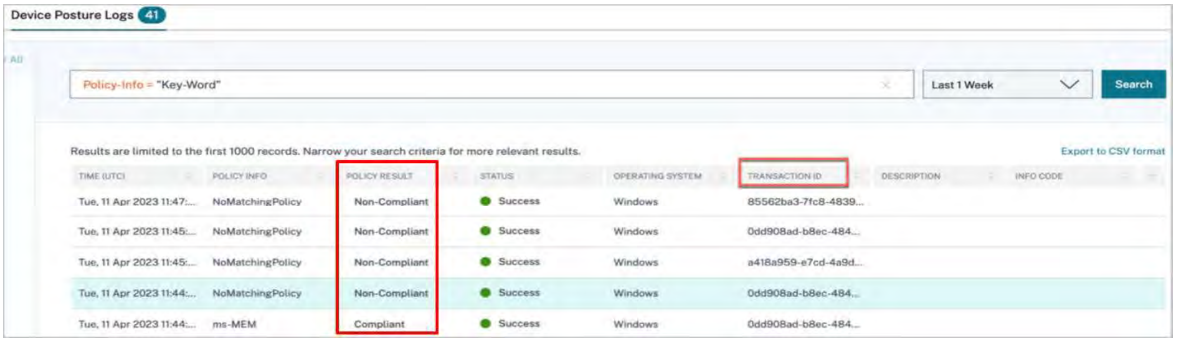
3. Continuous monitoring.

Granular visibility and the continuous monitoring of network traffic, user activities, and device behavior gives security teams heightened visibility in their environments. This empowers security teams to detect and respond to potential threats by promptly identifying anomalous behavior or malicious activities within the network.

Based on the evaluation of these attributes, Citrix adjusts access privileges based on real-time context and facilitates continuous monitoring of user behavior to detect anomalous activities.

Contextual and Adaptive Access Control: Citrix provides granular control over resource access based on contextual factors such as user identity, device posture, network, geolocation, and behavior. Citrix also integrates with 3rd party detection response and unified endpoint management solutions for device posture. By incorporating these contextual attributes into access control policies, organizations can enforce a zero-trust approach by dynamically granting or denying access to resources based on real-time risk assessments.

(*See* Zero Trust & Citrix, available at https://www.citrix.com/content/dam/citrix/en_us/documents/white-paper/zero-trust-and-citrix.pdf (last accessed April 23, 2025).)



(*See* Device Posture logs and events, available at https://docs.citrix.com/en-us/device-posture/device-posture-logs.html (published December 17, 2024) (last accessed April 23, 2025).) Accordingly, the Accused Instrumentalities meet limitation F of Claim 19.

56. Additionally, and/or alternatively, Citrix has indirectly infringed and continues to indirectly infringe one or more of the claims of the '705 Patent, in violation of 35 U.S.C. § 271(b) by actively inducing users of the SPA system and/or devices operating in the SPA ecosystem to directly infringe one or more claims of the '705 Patent. For example, (a) Citrix had actual knowledge of or was willfully blind to the existence of the '705 Patent no later than January 31, 2025, when it received the letter attached as Exhibit 5, and (b) Citrix intentionally causes, urges, or encourages users of the Accused Instrumentalities to take action that, when taken, directly infringe one or more claims of the '705 Patent. Citrix's encouragement is accomplished by promoting, advertising, and instructing customers and potential customers to use the Accused Instrumentalities and/or devices utilizing the Accused Instrumentalities, including infringing uses thereof. Citrix knows (based on the claim chart previously provided by K.Mizra and/or after reading this Complaint should know) that its actions will induce users of the SPA products and ecosystem to directly infringe one or more claims of the '705 Patent, and users thereof directly infringe one or more claims of the '705 Patent. For instance, at a minimum, Citrix has supplied and

continues to supply the Accused Instrumentalities to customers while knowing that installation and use thereof will infringe one or more claims of the '705 Patent.

57. Citrix's acts of infringement have occurred within this District and elsewhere throughout the United States.

58. As a result of Citrix's infringing conduct, K.Mizra has suffered damages. Citrix is liable to K.Mizra in an amount that adequately compensates K.Mizra for Citrix's infringement in an amount that is no less than a fully paid-up, lump-sum, reasonable royalty, together with interest and costs as fixed by this Court under 25 U.S.C. § 284.

REQUEST FOR RELIEF

WHEREFORE, K.Mizra respectfully requests the Court find in its favor and against Citrix, and that the Court grant K.Mizra at least the following relief:

A. Judgment that Citrix has directly infringed, literally and/or under the DoE, one or more claims of the Asserted Patent;

B. Awarding damages to K.Mizra in an amount to be proven at trial and in the form of a fully paid-up, lump sum, reasonable royalty that takes into account and runs through expiration of the Asserted Patents;

C. Awarding enhanced damages, as appropriate, under 35 U.S.C. § 284;

D. Awarding K.Mizra's costs (including disbursements) and declaring this an exceptional case and awarding K.Mizra its attorneys' fees in accordance with 35 U.S.C. § 285;

E. Pre-judgment and post-judgment interest at the maximum rate permitted by law on the damages caused to K.Mizra by reason of Citrix's infringing activities and other conduct complaint of herein; and

K.Mizra LLC, Ex. 2001, p. 29
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

F. Awarding such other and further relief as this Court deems just and proper under the circumstances.

DEMAND FOR JURY TRIAL

Pursuant to Fed. R. Civ. P. 38(b), K.Mizra hereby demands a trial by jury on all issues so triable.

Dated: April 24, 2025

Respectfully Submitted,

/s/ Gerald E. Greenberg

GERALD E. GREENBERG
Florida Bar No. 440094
ggreenberg@gsgpa.com
ALESSANDRA M. SIBLESZ
Florida Bar No. 1024843
asiblesz@gsgpa.com
GELBER SCHACHTER & GREENBERG, P.A.
One Southeast Third Avenue, Suite 2600
Miami, Florida 33131
Telephone: (305) 728-0950
E-service: efilings@gsgpa.com

ROBERT R. BRUNELLI*
CO State Bar No. 20070
rbrunelli@sheridanross.com
BRIAN S. BOERMAN*
CO State Bar No. 50834
bboerman@sheridanross.com
GENE VOLCHENKO*
IL State Bar No. 6342818
gvolchenko@sheridanross.com
SHERIDAN ROSS P.C.
1560 Broadway, Suite 1200
Denver, CO 80202
Telephone: 303-863-9700
E-service: litigation@sheridanross.com

**To be admitted pro hac vice*

Counsel for Plaintiff K.Mizra LLC

K.Mizra LLC, Ex. 2001, p. 30
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

EXHIBIT 1

citrix

Contact us

Support Sales Partners Services

Headquarters

Corporate headquarters

851 Cypress Creek Road
Fort Lauderdale, FL 33309
United States
Map and directions

Toll free: 1 800 424 8749
Phone: (954) 267 3000

K.Mizra LLC, Ex. 2001, p. 32
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468



Regional headquarters

© 2025 Cloud Software Group, Inc. All rights reserved.

EXHIBIT 2

Detail by Entity Name

Foreign Profit Corporation
CLOUD SOFTWARE GROUP, INC.

Filing Information

Document Number	F00000004765
FEI/EIN Number	77-0449727
Date Filed	08/23/2000
State	DE
Status	ACTIVE
Last Event	AMENDMENT AND NAME CHANGE
Event Date Filed	12/02/2022
Event Effective Date	NONE

Principal Address

851 W. Cypress Creek Road
Fort Lauderdale, FL 33309

Changed: 04/09/2024

Mailing Address

851 W. Cypress Creek Road
Fort Lauderdale, FL 33309

Changed: 04/09/2024

Registered Agent Name & Address

CT CORPORATION SYSTEM
1200 SOUTH PINE ISLAND ROAD
PLANTATION, FL 33324

Name Changed: 05/13/2024

Officer/Director Detail

Name & Address

Title Treasurer / CFO

Berquist, Thomas
851 W. Cypress Creek Road
Fort Lauderdale, FL 33309

Title Director

Smith, Robert F
851 W. Cypress Creek Road
Fort Lauderdale, FL 33309

Title Director

Saroya, Maneet S.
851 W. Cypress Creek Road
Fort Lauderdale, FL 33309

Title Director

Hung, Betty
851 W. Cypress Creek Road
Fort Lauderdale, FL 33309

Title Assistant Secretary

Stalder, John
851 W. Cypress Creek Road
Fort Lauderdale, FL 33309

Title Director

Stalder, John
851 W. Cypress Creek Road
Fort Lauderdale, FL 33309

Title President / CEO/ Director

Krause, Thomas H.
851 W. Cypress Creek Road
Fort Lauderdale, FL 33309

Title Chairman of the Board

Krause, Thomas H.
851 W. Cypress Creek Road
Fort Lauderdale, FL 33309

Title Secretary / CLO

Gomes, Antonio
851 W. Cypress Creek Road
Fort Lauderdale, FL 33309

Title Director

White, Steven
851 W. Cypress Creek Road
Fort Lauderdale, FL 33309

Title Director

Cohn, Jesse
851 W. Cypress Creek Road
Fort Lauderdale, FL 33309

Title Director

Genrich, Jason
851 W. Cypress Creek Road
Fort Lauderdale, FL 33309

Title Director

Hsu, Chris
851 W. Cypress Creek Road
Fort Lauderdale, FL 33309

Title Assistant Treasurer

Shytle, Brian L.
851 W. Cypress Creek Road
Fort Lauderdale, FL 33309

Title Director

Croll, Robert
851 W. Cypress Creek Road
Fort Lauderdale, FL 33309

Title Assistant Secretary

Croll, Robert
851 W. Cypress Creek Road
Fort Lauderdale, FL 33309

Title Director

Kerko, David
851 W. Cypress Creek Road
Fort Lauderdale, FL 33309

Annual Reports

Report Year	Filed Date
2024	04/09/2024
2024	05/13/2024
2025	02/14/2025

K.Mizra LLC, Ex. 2001, p. 38
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

Document Images

02/14/2025 -- ANNUAL REPORT	View image in PDF format
05/13/2024 -- AMENDED ANNUAL REPORT	View image in PDF format
04/09/2024 -- ANNUAL REPORT	View image in PDF format
03/08/2023 -- ANNUAL REPORT	View image in PDF format
12/02/2022 -- Amendment and Name Change	View image in PDF format
03/31/2022 -- ANNUAL REPORT	View image in PDF format
04/22/2021 -- ANNUAL REPORT	View image in PDF format
05/31/2020 -- ANNUAL REPORT	View image in PDF format
03/20/2019 -- ANNUAL REPORT	View image in PDF format
04/07/2018 -- ANNUAL REPORT	View image in PDF format
07/10/2017 -- ANNUAL REPORT	View image in PDF format
11/01/2016 -- REINSTATEMENT	View image in PDF format
05/05/2015 -- ANNUAL REPORT	View image in PDF format
05/01/2014 -- ANNUAL REPORT	View image in PDF format
04/05/2013 -- ANNUAL REPORT	View image in PDF format
04/25/2012 -- ANNUAL REPORT	View image in PDF format
04/26/2011 -- ANNUAL REPORT	View image in PDF format
06/18/2010 -- ANNUAL REPORT	View image in PDF format
04/24/2009 -- ANNUAL REPORT	View image in PDF format
05/05/2008 -- ANNUAL REPORT	View image in PDF format
05/07/2007 -- ANNUAL REPORT	View image in PDF format
05/02/2006 -- ANNUAL REPORT	View image in PDF format
09/02/2005 -- ANNUAL REPORT	View image in PDF format
07/19/2004 -- ANNUAL REPORT	View image in PDF format
08/01/2003 -- ANNUAL REPORT	View image in PDF format
08/25/2002 -- ANNUAL REPORT	View image in PDF format
03/14/2001 -- ANNUAL REPORT	View image in PDF format
08/23/2000 -- Foreign Profit	View image in PDF format

Florida Department of State, Division of Corporations

EXHIBIT 3

505861362 01/13/2020

PATENT ASSIGNMENT COVER SHEET

Electronic Version v1.1
Stylesheet Version v1.2

EPAS ID: PAT5908299

SUBMISSION TYPE:	NEW ASSIGNMENT
NATURE OF CONVEYANCE:	ASSIGNMENT

CONVEYING PARTY DATA

Name	Execution Date
NETWORK SECURITY TECHNOLOGIES, LLC	12/24/2019

RECEIVING PARTY DATA

Name:	K.MIZRA LLC
Street Address:	2160 CENTURY PARK EAST
Internal Address:	#707
City:	LOS ANGELES
State/Country:	CALIFORNIA
Postal Code:	90067

PROPERTY NUMBERS Total: 5

Property Type	Number
Patent Number:	8234705
Patent Number:	9398037
Patent Number:	7630381
Patent Number:	8149844
Patent Number:	9516048

CORRESPONDENCE DATA

Fax Number: (212)586-5095

Correspondence will be sent to the e-mail address first; if that is unsuccessful, it will be sent using a fax number, if provided; if that is unsuccessful, it will be sent via US Mail.

Phone: 2125865800
Email: DFRIEDMAN@CTSWLAW.COM
Correspondent Name: DANIEL J. FRIEDMAN
Address Line 1: 420 LEXINGTON AVENUE
Address Line 2: SUITE 2400
Address Line 4: NEW YORK, NEW YORK 10170

NAME OF SUBMITTER:	DANIEL J. FRIEDMAN
SIGNATURE:	/Daniel J. Friedman/
DATE SIGNED:	01/13/2020
	This document serves as an Oath/Declaration (37 CFR 1.63).

Total Attachments: 2
source=00407807#page1.tif
source=00407807#page2.tif

DocuSign Envelope ID: FC81257A-F6D0-4296-89B4-7D6A643A611E

ASSIGNMENT OF PATENTS AND PATENT RIGHTS

This ASSIGNMENT OF PATENTS AND PATENT RIGHTS (the "Assignment") is executed, acknowledged and delivered by Network Security Technologies, LLC, a Delaware limited liability company ("Assignor").

NOW, THEREFORE, TO ALL WHOM IT MAY CONCERN:

For good and valuable consideration, the receipt of which is hereby acknowledged, Assignor agrees to and does hereby irrevocably sell, assign, transfer, and convey unto K.Mizra LLC, a Delaware limited liability company, with its principal place of business at 2160 Century Park East # 707, Los Angeles, CA 90067 ("Assignee"), and Assignee hereby accepts, all of Assignor's right, title, and interest (i) in and to the patents and patent rights set forth on Exhibit 1 hereto (the "Patents"), together with the goodwill associated therewith, the same to be held and enjoyed by said Assignee for its own use, and for the use of its successors, assigns, or other legal representatives to the end of the term or terms for which said Patents may be granted as fully and entirely as the same would have been held and enjoyed by Assignor if this Assignment had not been made; and (ii) in and to any and all causes of action and enforcement rights for the Patents including, without limitation, all rights to pursue damages, injunctive relief and other remedies for past and future infringement of the Patents.

This Assignment shall be binding upon and shall inure to the benefit of the parties and their respective successors and assigns.

Assignee is hereby authorized to file or record this Assignment or any other instrument in such public offices and with such governmental authorities, including, without limitation, the United States Patent and Trademark Office, as Assignee may determine from time to time for the purpose of evidencing the foregoing assignment.

This Assignment shall be governed by, and construed in accordance with, the laws of the United States in respect to patent issues and in all other respects by the laws of the State of New York, without giving effect to the conflict of laws rules thereof.

IN WITNESS WHEREOF, Assignor has caused this Assignment to be executed as of this 24th day of December 2019.

Network Security Technologies, LLC

DocuSigned by:
By: _____
DAAEA6F409B34F9...
Name: Brian Yates
Title: CEO

DocuSign Envelope ID: FC81257A-F6D0-4296-89B4-7D6A643A611E

EXHIBIT 1
Patents

Owner	Country	Application No.	Patent No.
Network Security Technologies, LLC	United States	11/237,003	8234705
Network Security Technologies, LLC	United States	11/237,004	9398037
Network Security Technologies, LLC	United States	11/237,199	7630381
Network Security Technologies, LLC	United States	12/605,338	8149844
Network Security Technologies, LLC	United States	15/206,227	9516048

EXHIBIT 4

US008234705B1

(12) **United States Patent**
Roskind et al.

(10) **Patent No.: US 8,234,705 B1**
(45) **Date of Patent: Jul. 31, 2012**

(54) **CONTAGION ISOLATION AND INOCULATION**

(75) Inventors: **James A. Roskind**, Redwood City, CA (US); **Aaron T. Emigh**, Incline Village, NV (US)

(73) Assignee: **Radix Holdings, LLC**, Incline Village, NV (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 1520 days.

(21) Appl. No.: **11/237,003**

(22) Filed: **Sep. 27, 2005**

Related U.S. Application Data

(60) Provisional application No. 60/613,909, filed on Sep. 27, 2004.

(51) **Int. Cl.**
G06F 11/00 (2006.01)
G06F 12/14 (2006.01)
G06F 12/16 (2006.01)
G08B 23/00 (2006.01)

(52) **U.S. Cl.** **726/22**; 726/23; 726/24; 726/25

(58) **Field of Classification Search** 726/11, 726/14, 22–25

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,823,479	B1 *	11/2004	McElhaney et al.	714/43
6,996,845	B1 *	2/2006	Hurst et al.	726/25
7,263,720	B2 *	8/2007	Silvester	726/22
7,287,278	B2 *	10/2007	Liang	726/22
7,424,610	B2 *	9/2008	Ranganathan	713/164
2002/0083328	A1 *	6/2002	Riordan	713/183
2003/0188173	A1 *	10/2003	Zimmer et al.	713/189
2003/0188194	A1 *	10/2003	Currie et al.	713/201
2005/0018618	A1 *	1/2005	Mualem et al.	370/252
2005/0033987	A1 *	2/2005	Yan et al.	713/201
2005/0131997	A1 *	6/2005	Lewis et al.	709/203
2006/0053426	A1 *	3/2006	Dive-Reclus et al.	719/328

OTHER PUBLICATIONS

Bohra, A.; Neamtiu, I.; Gallard, P.; Sultan, F.; Iftode, L.; Remote repair of operating system state using Backdoors; Autonomic Computing, 2004. Proceedings. International Conference on May 17-18, 2004 pp. 256-263.*

Mark V. Glover, Integrating a Trusted Computing Base Extension Server and Secure Session Server into the LINUX Operating System, Sep. 2001, Naval Postgraduate School.*

Pearson et al., Trusted Computing Platform: TCPA Technology in Context, 2003, Hewlett-Packard Books: Patricia Pekary.*

OLS: Linux and trusted computing, Jul. 22, 2005, http://lwn.net/Articles/144681/ (as lined on Jul. 15, 2010).*

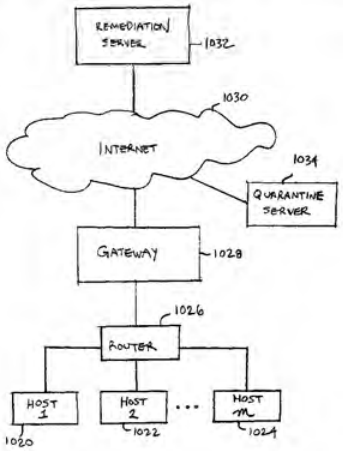
* cited by examiner

Primary Examiner — Eleni Shiferaw
Assistant Examiner — Jing Sims

(57) **ABSTRACT**

Contagion isolation and inoculation is disclosed. In some embodiments, a request is received from a host, e.g., via a network interface, to connect to a protected network. It is determined whether the host is required to be quarantined. If the host is required to be quarantined, the host is provided only limited access to the protected network. In some embodiments, a quarantined host is permitted to access the protected network only as required to remedy a condition that caused the quarantine to be imposed, such as to download a software patch, update, or definition; install, remove, and/or configure software and/or settings as required by a policy; and/or to have a scan or other diagnostic and/or remedial operation performed. In various embodiments attempts to communicate with hosts not involved in remediation are redirected to a quarantine system, such as a server, that provides information, notices, updates, and/or instructions to the user.

19 Claims, 25 Drawing Sheets



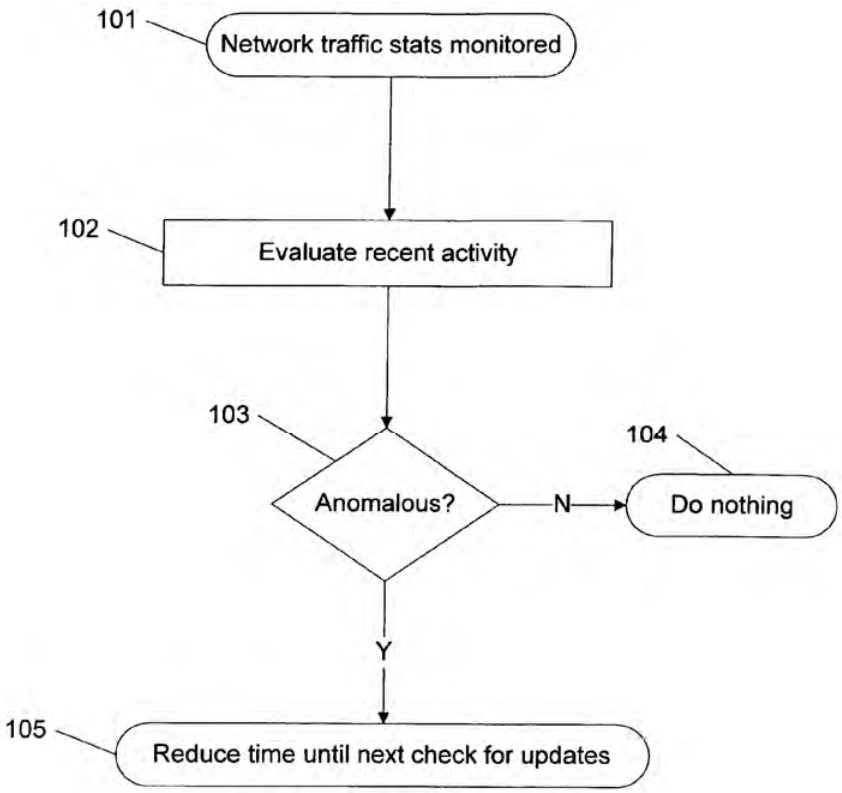

101
Network traffic stats monitored
102
Evaluate recent activity
103
Anomalous?
N
104
Do nothing
Y
105
Reduce time until next check for updates

Figure 1

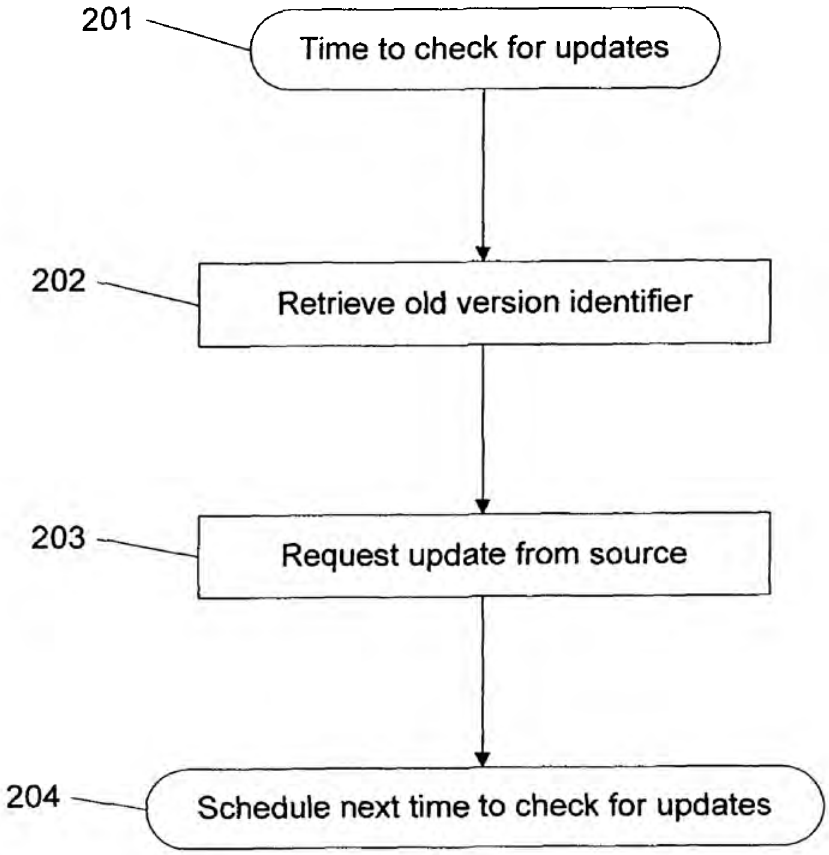

201
Time to check for updates
202
Retrieve old version identifier
203
Request update from source
204
Schedule next time to check for updates

Figure 2A

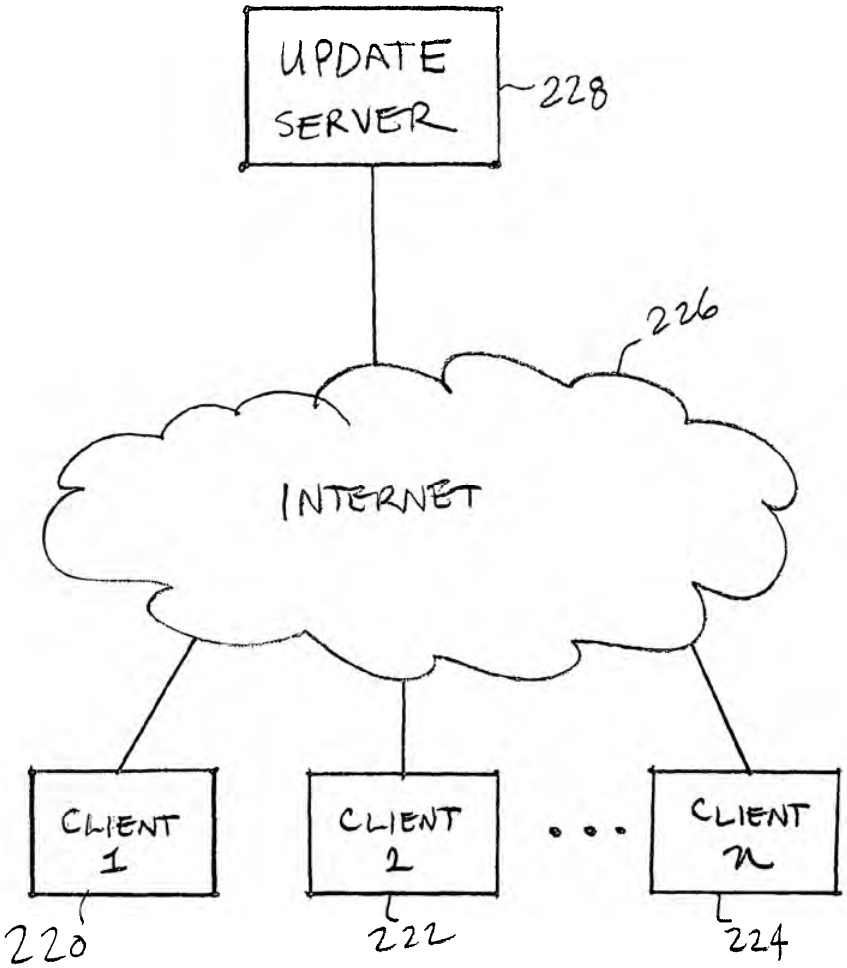

UPDATE SERVER
228
226
INTERNET
CLIENT 1
CLIENT 2
CLIENT n
220
222
224

FIG. 2B

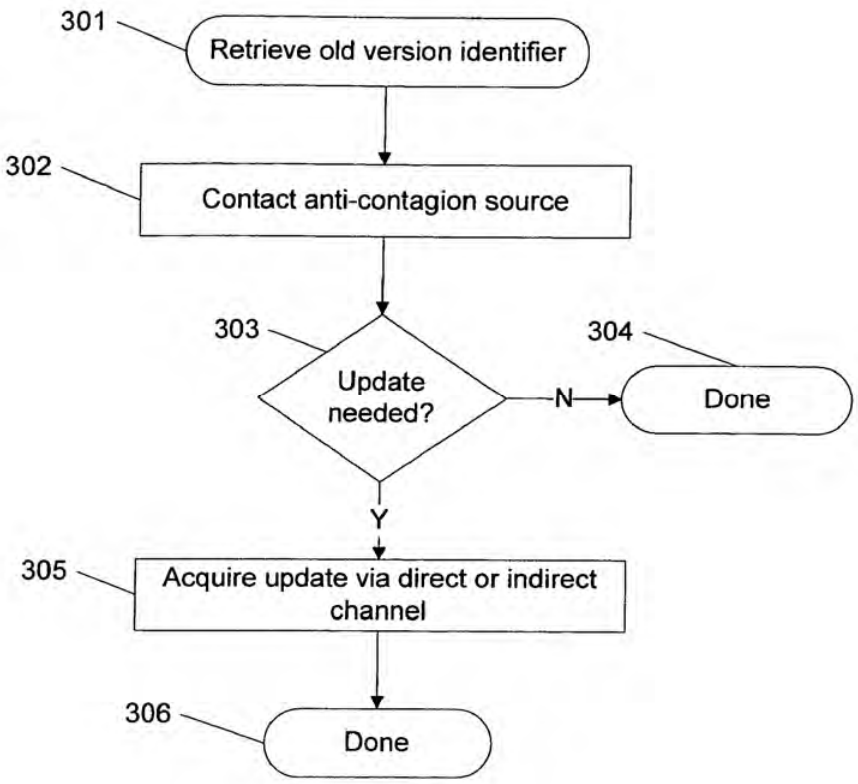

301
Retrieve old version identifier
302
Contact anti-contagion source
303
Update needed?
304
N
Done
Y
305
Acquire update via direct or indirect channel
306
Done

Figure 3A

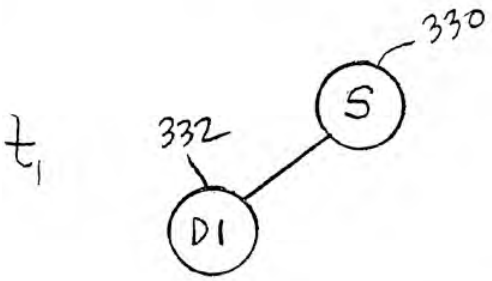


FIG. 3B

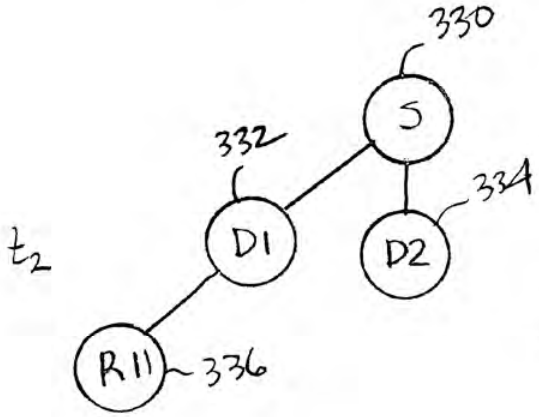


FIG. 3C

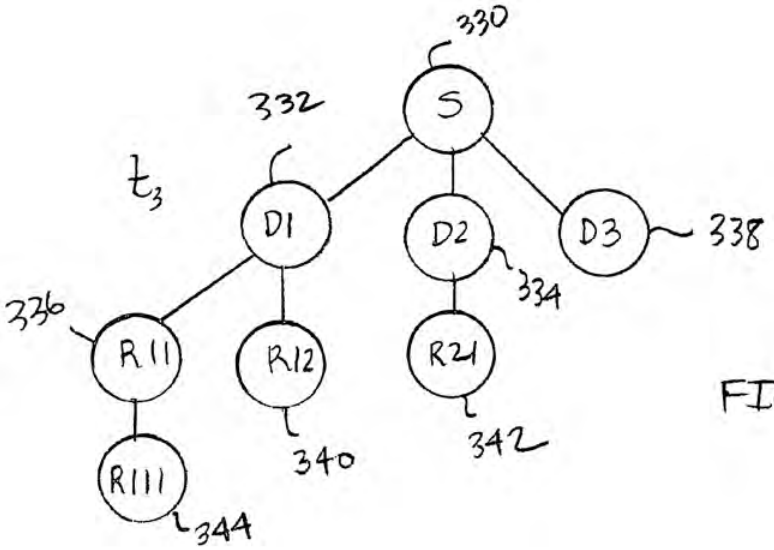


FIG. 3D

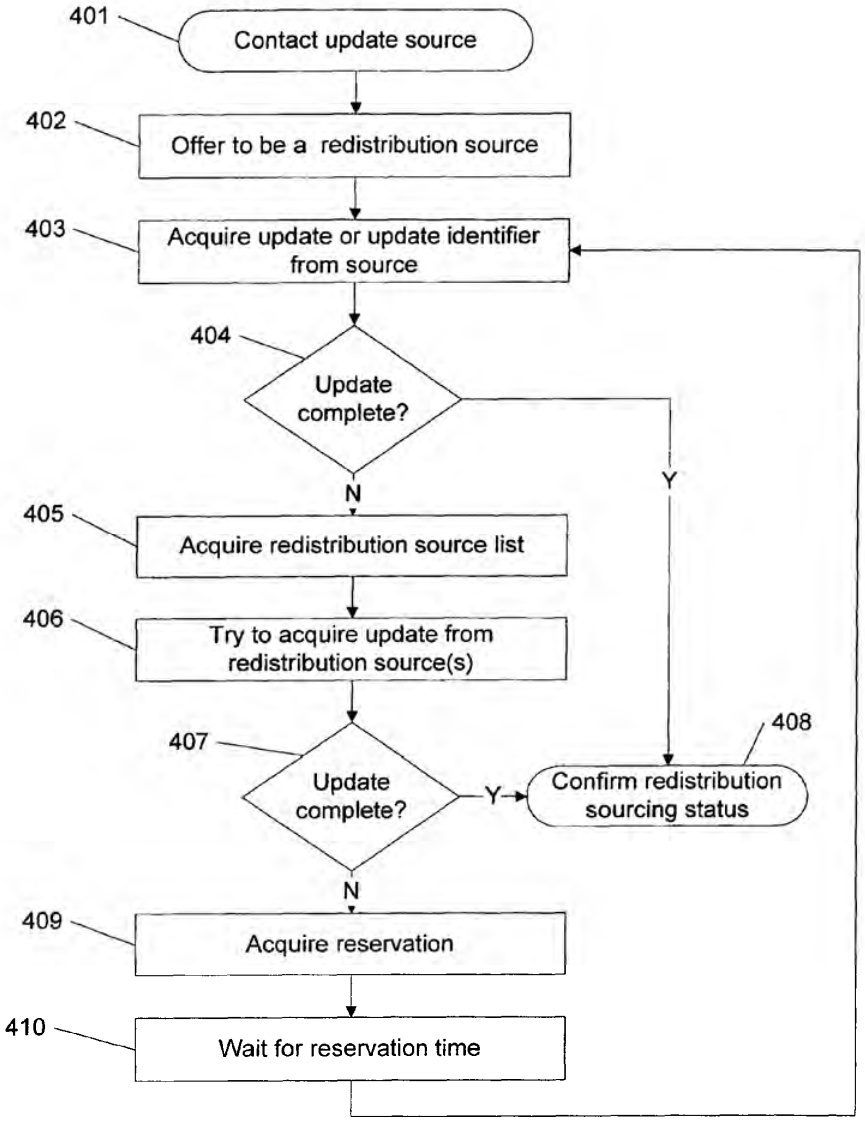

401
Contact update source
402
Offer to be a redistribution source
403
Acquire update or update identifier from source
404
Update complete?
N
Y
405
Acquire redistribution source list
406
Try to acquire update from redistribution source(s)
407
Update complete?
Y
408
Confirm redistribution sourcing status
N
409
Acquire reservation
410
Wait for reservation time

Figure 4A

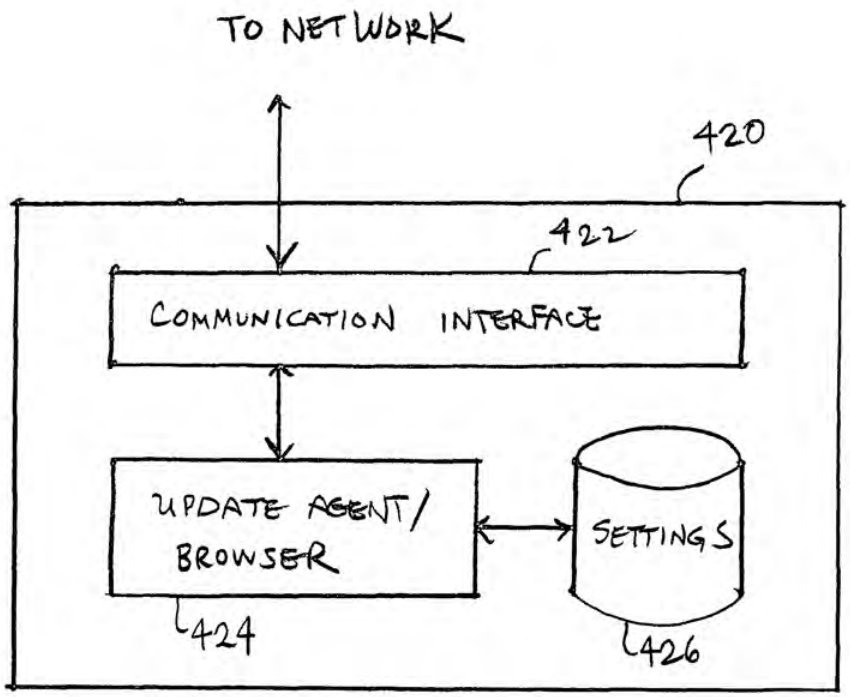

TO NETWORK
420
422
COMMUNICATION INTERFACE
UPDATE AGENT/
BROWSER
424
SETTINGS
426

FIG. 4B

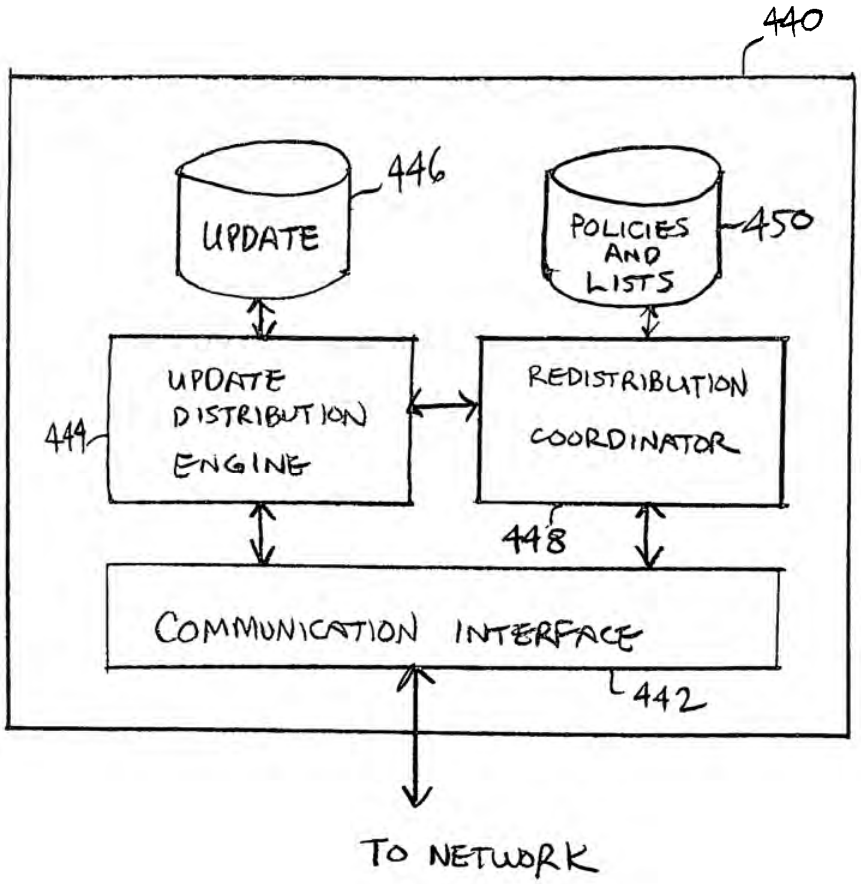

440
446
UPDATE
POLICIES AND LISTS
450
UPDATE DISTRIBUTION ENGINE
444
REDISTRIBUTION COORDINATOR
448
COMMUNICATION INTERFACE
442
TO NETWORK

FIG. 4C

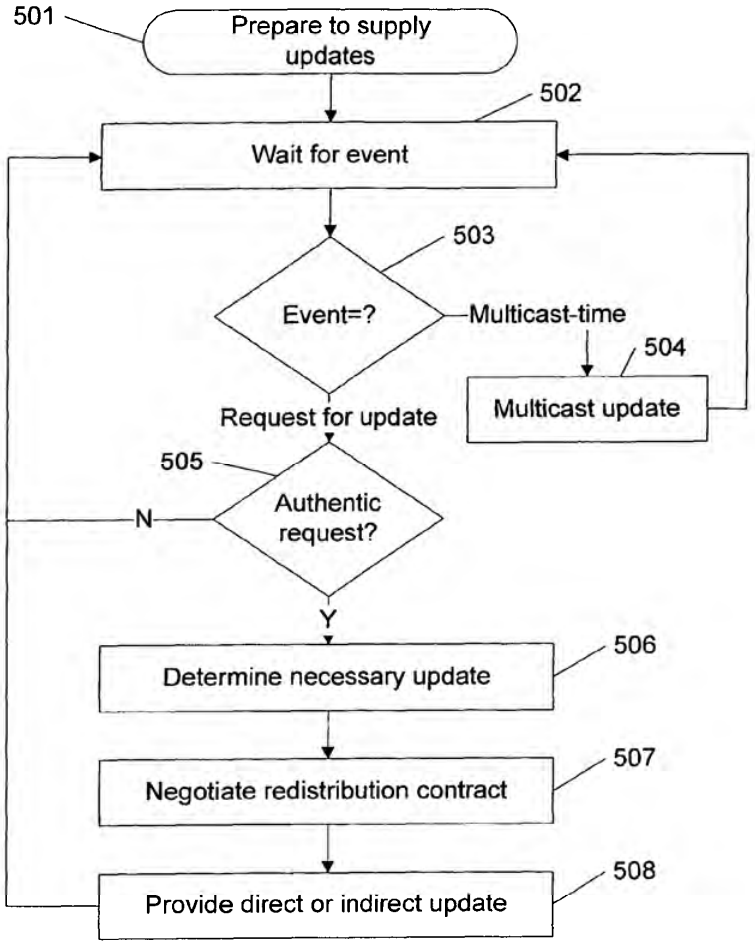

501
Prepare to supply updates
502
Wait for event
503
Event=?
Multicast-time
504
Multicast update
Request for update
505
Authentic request?
N
Y
506
Determine necessary update
507
Negotiate redistribution contract
508
Provide direct or indirect update

Figure 5

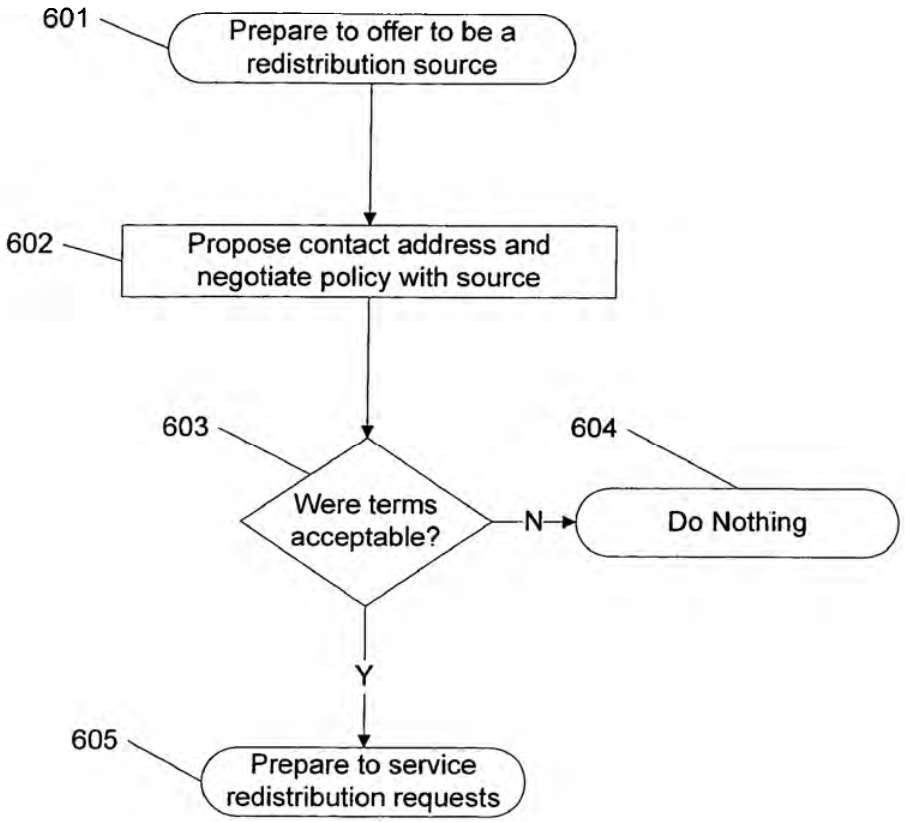

601
Prepare to offer to be a redistribution source
602
Propose contact address and negotiate policy with source
603
Were terms acceptable?
N
604
Do Nothing
Y
605
Prepare to service redistribution requests

Figure 6

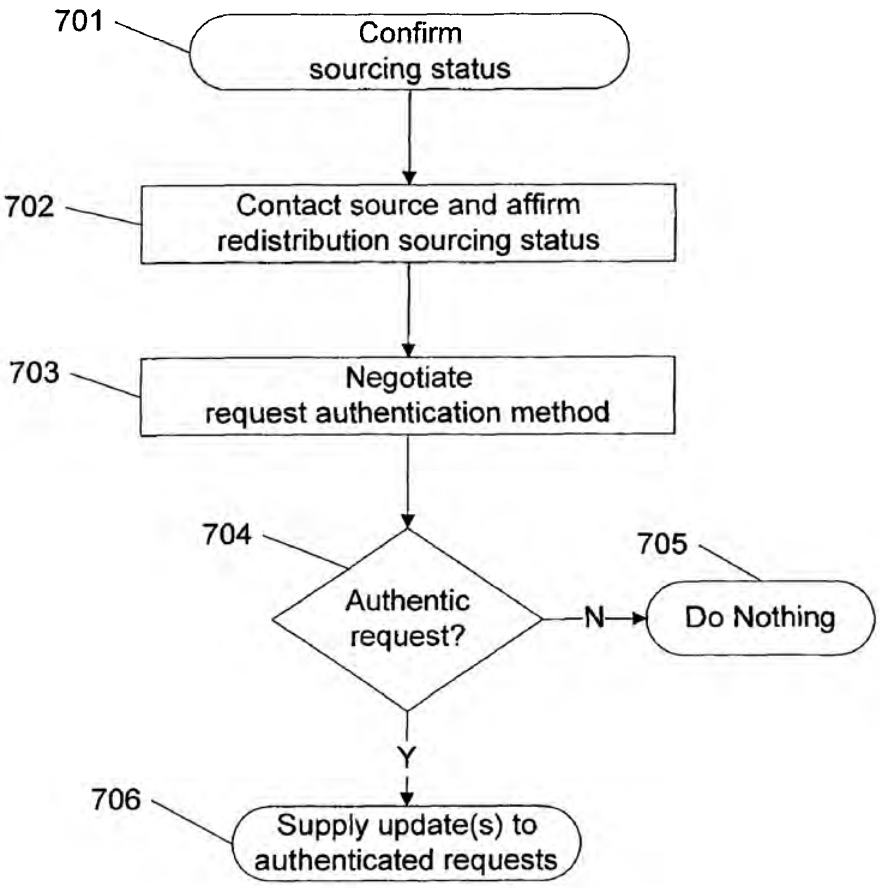

701
Confirm sourcing status
702
Contact source and affirm redistribution sourcing status
703
Negotiate request authentication method
704
Authentic request?
N
705
Do Nothing
Y
706
Supply update(s) to authenticated requests

Figure 7

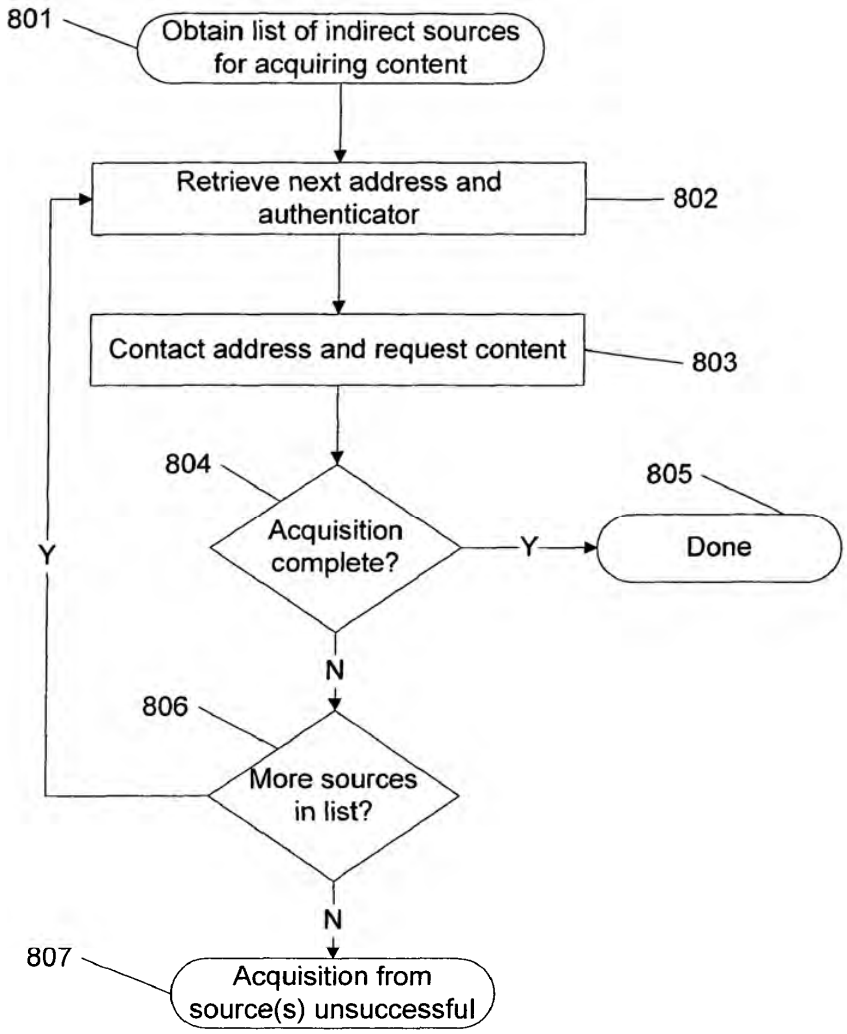

801
Obtain list of indirect sources for acquiring content
Retrieve next address and authenticator
802
Contact address and request content
803
804
Acquisition complete?
805
Done
Y
Y
N
806
More sources in list?
Y
N
807
Acquisition from source(s) unsuccessful

Figure 8

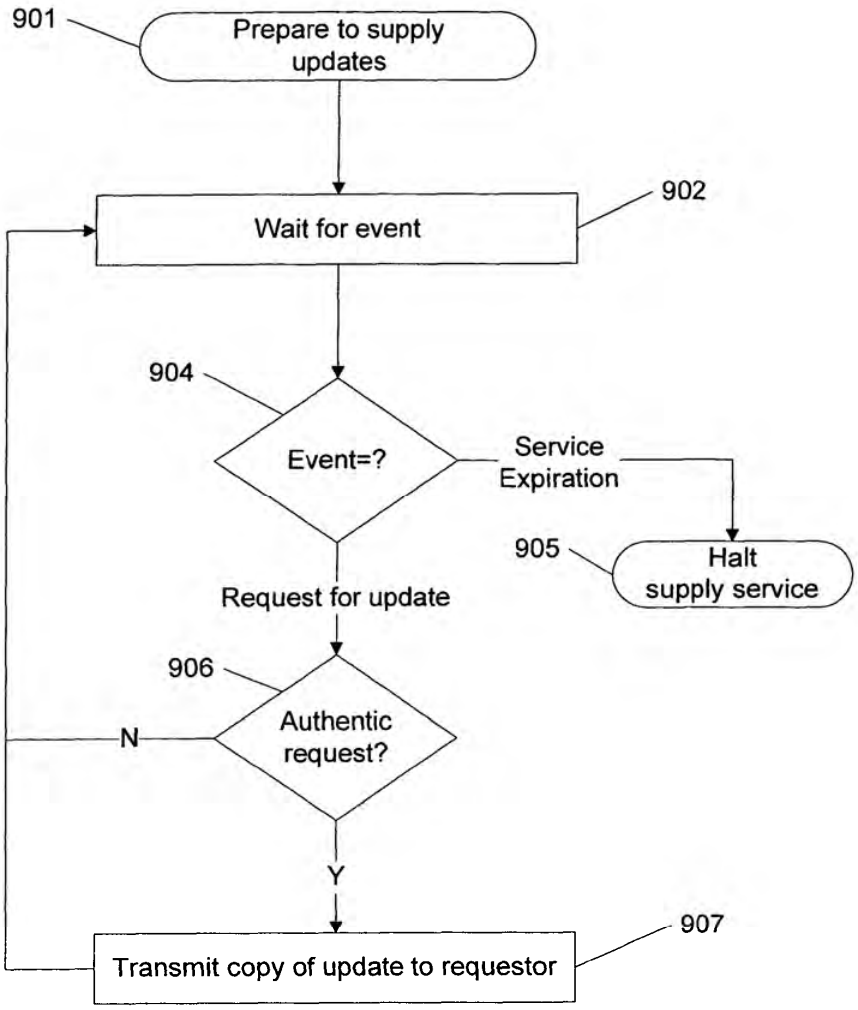

901
Prepare to supply updates
902
Wait for event
904
Event=?
Service Expiration
905
Halt supply service
Request for update
906
Authentic request?
N
Y
907
Transmit copy of update to requestor

Figure 9

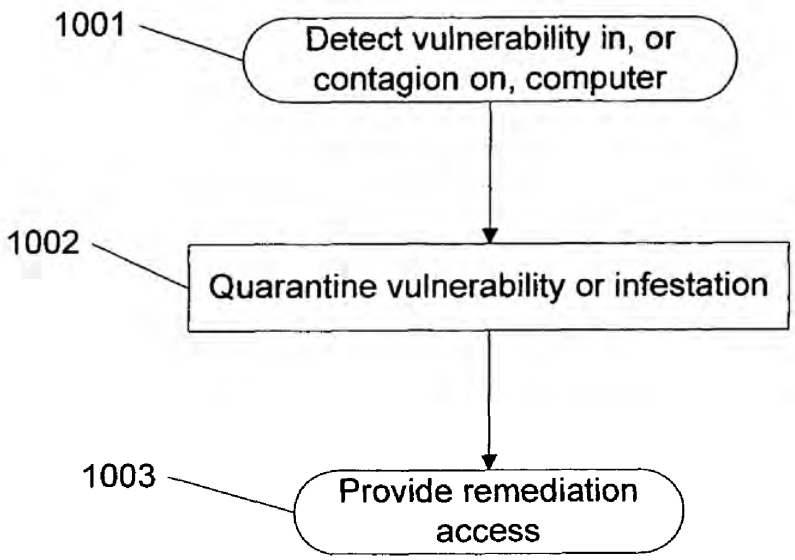

1001
Detect vulnerability in, or contagion on, computer
1002
Quarantine vulnerability or infestation
1003
Provide remediation access

Figure 10A

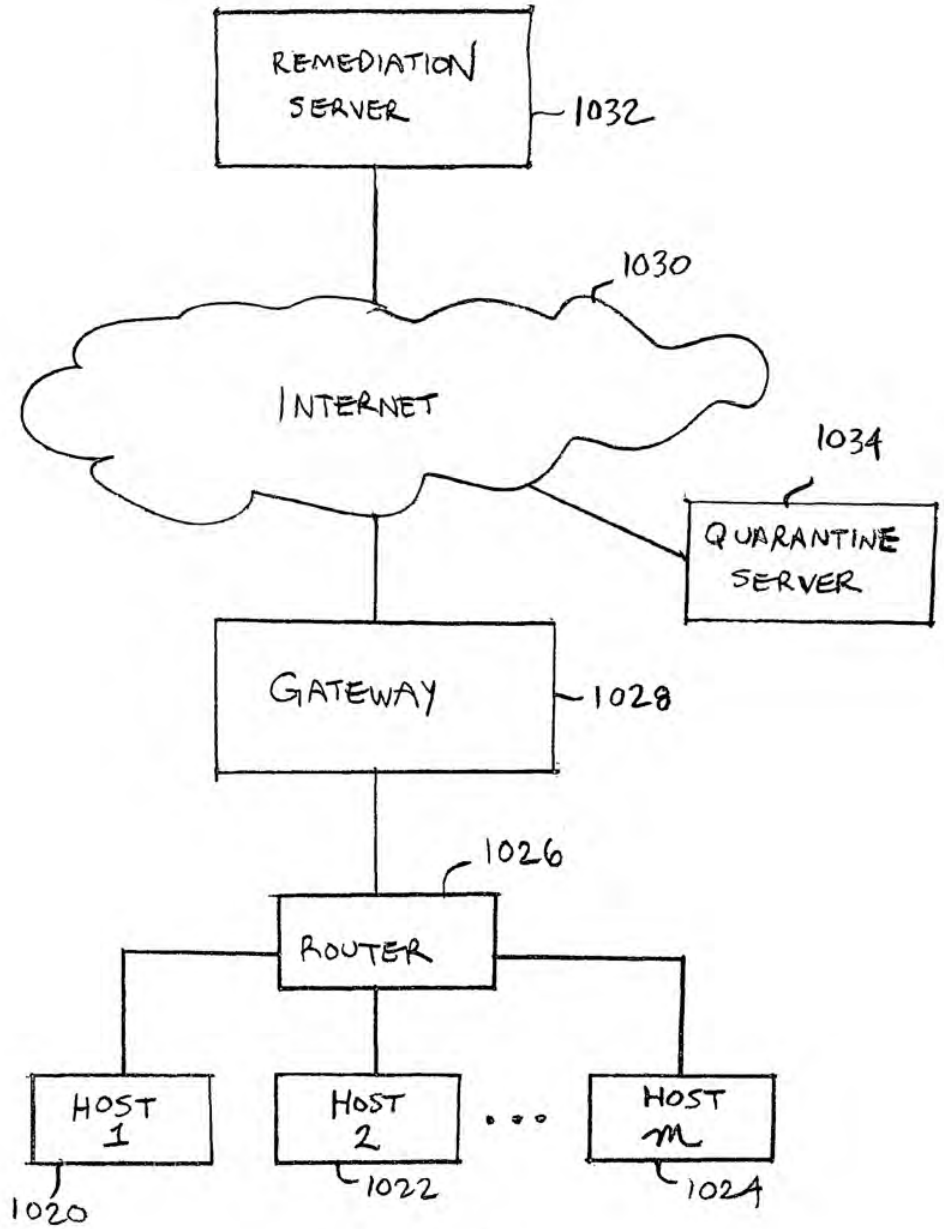

REMEDIATION SERVER
1032
1030
INTERNET
1034
QUARANTINE SERVER
GATEWAY
1028
1026
ROUTER
HOST 1
HOST 2
HOST m
1020
1022
1024

FIG. 10B

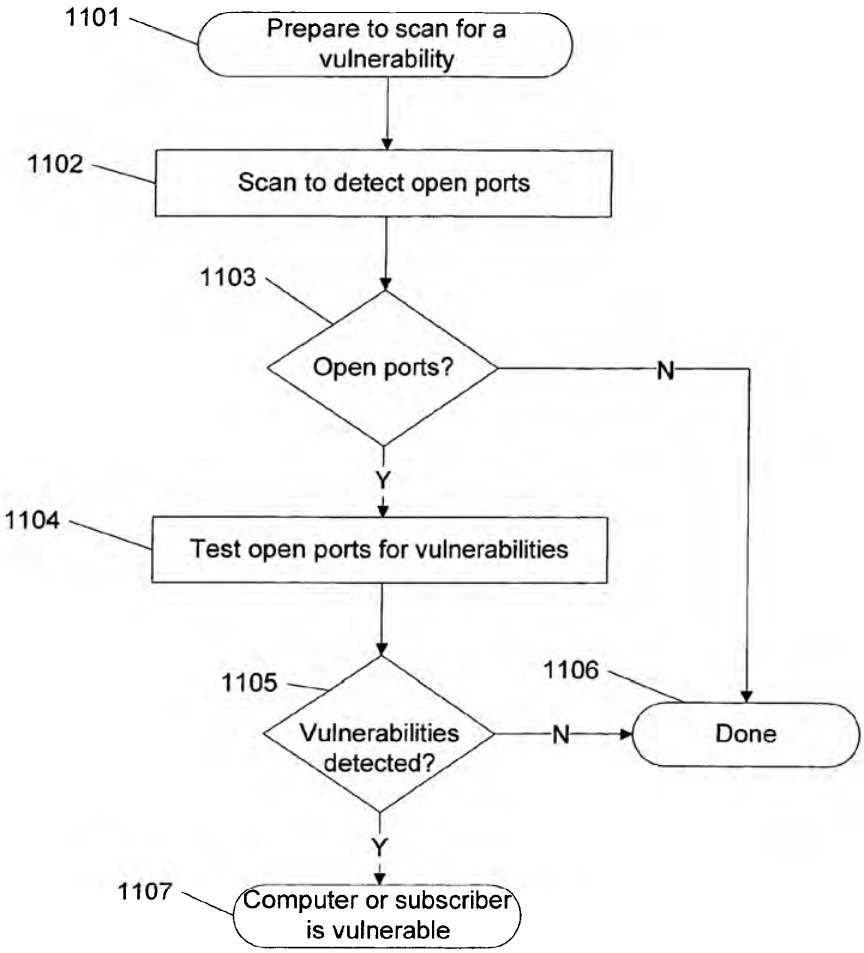

1101
Prepare to scan for a vulnerability
1102
Scan to detect open ports
1103
Open ports?
N
Y
1104
Test open ports for vulnerabilities
1105
Vulnerabilities detected?
N
1106
Done
Y
1107
Computer or subscriber is vulnerable

Figure 11

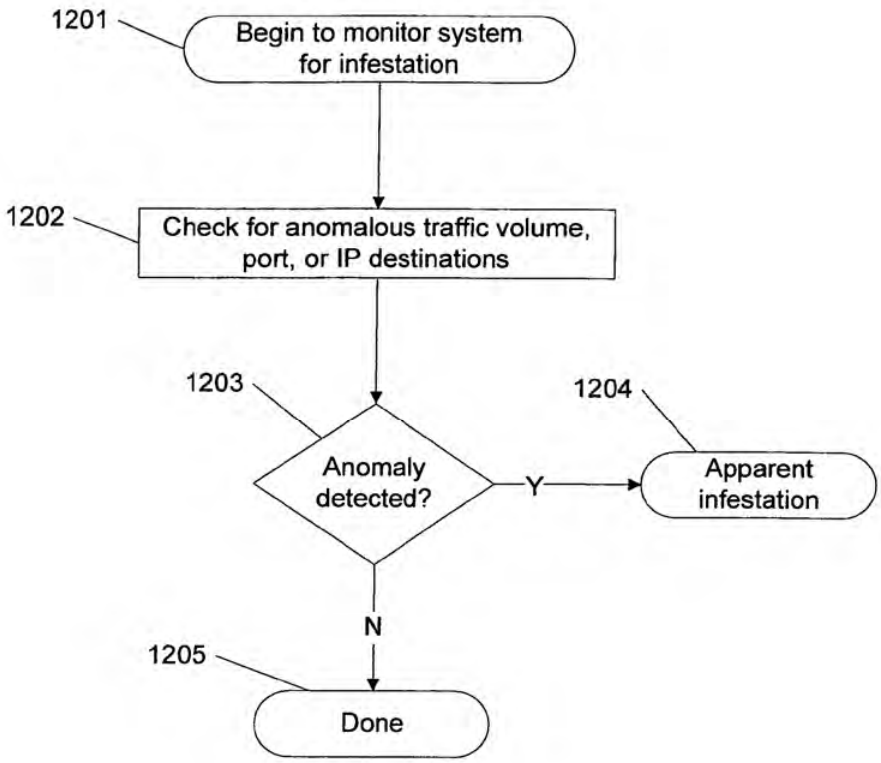

1201
Begin to monitor system for infestation
1202
Check for anomalous traffic volume, port, or IP destinations
1203
Anomaly detected?
Y
1204
Apparent infestation
N
1205
Done

Figure 12

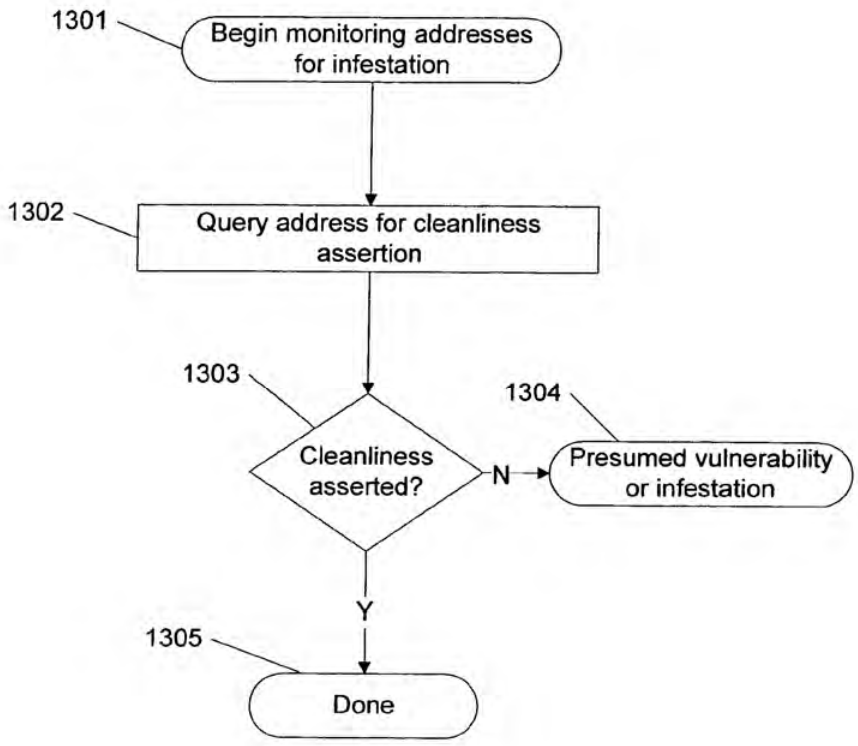

1301
Begin monitoring addresses for infestation
1302
Query address for cleanliness assertion
1303
Cleanliness asserted?
N
1304
Presumed vulnerability or infestation
Y
1305
Done

Figure 13

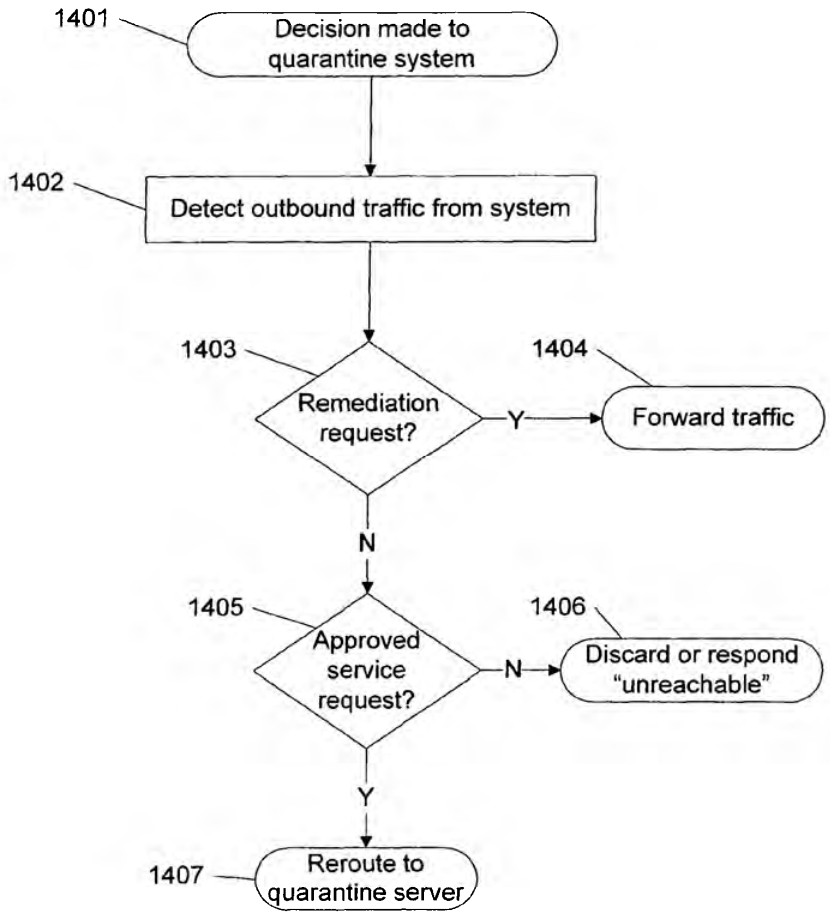

1401
Decision made to quarantine system
1402
Detect outbound traffic from system
1403
Remediation request?
Y
1404
Forward traffic
N
1405
Approved service request?
N
1406
Discard or respond "unreachable"
Y
1407
Reroute to quarantine server

Figure 14

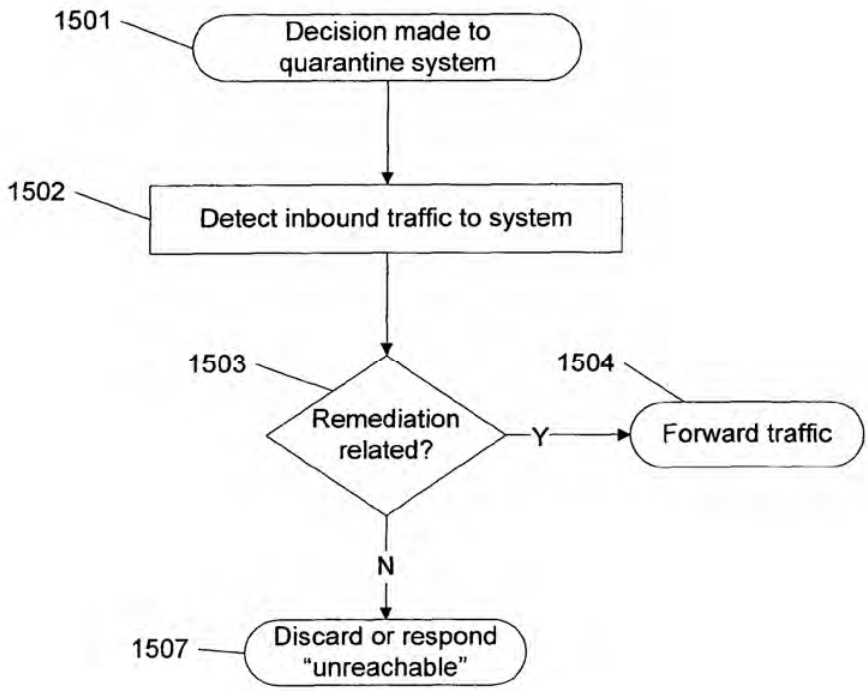

1501
Decision made to quarantine system
1502
Detect inbound traffic to system
1503
Remediation related?
Y
1504
Forward traffic
N
1507
Discard or respond "unreachable"

Figure 15

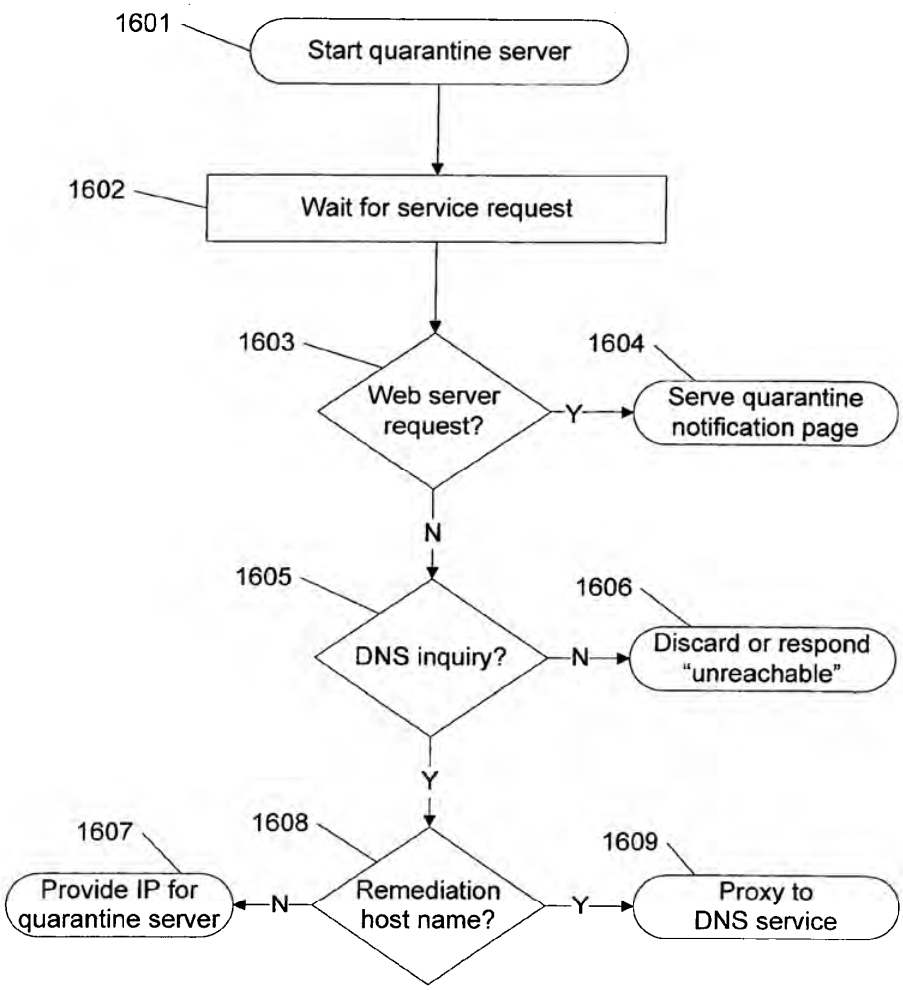

1601
Start quarantine server
1602
Wait for service request
1603
Web server request?
Y
1604
Serve quarantine notification page
N
1605
DNS inquiry?
N
1606
Discard or respond "unreachable"
Y
1607
Provide IP for quarantine server
N
1608
Remediation host name?
Y
1609
Proxy to DNS service

Figure 16

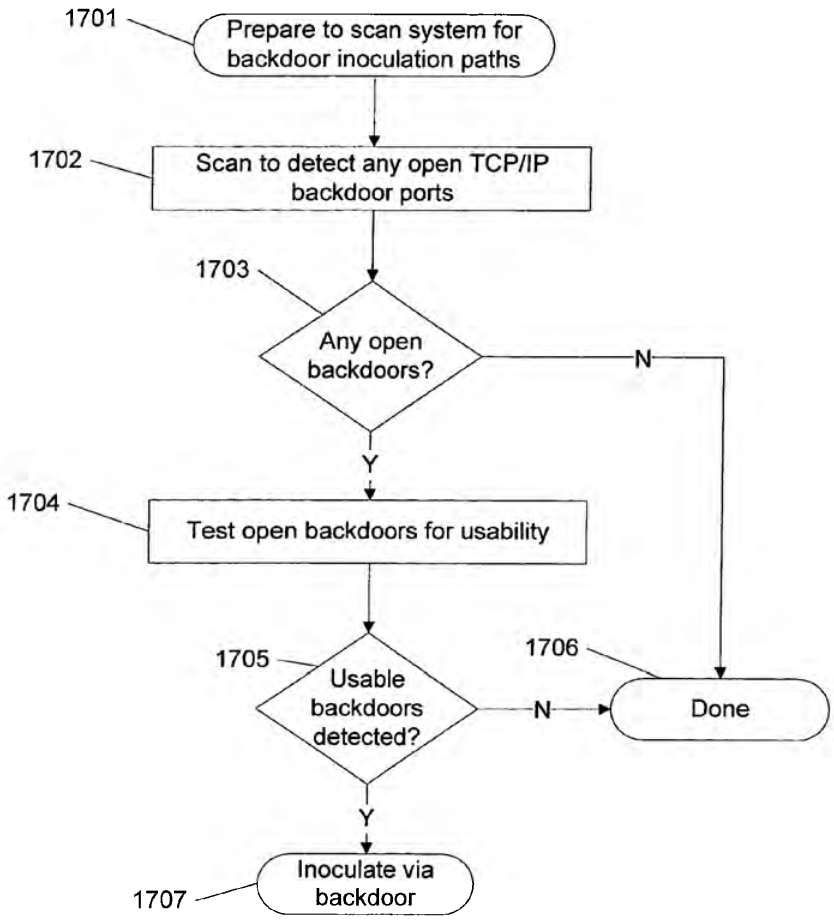

1701
Prepare to scan system for backdoor inoculation paths
1702
Scan to detect any open TCP/IP backdoor ports
1703
Any open backdoors?
N
Y
1704
Test open backdoors for usability
1705
Usable backdoors detected?
N
1706
Done
Y
1707
Inoculate via backdoor

Figure 17

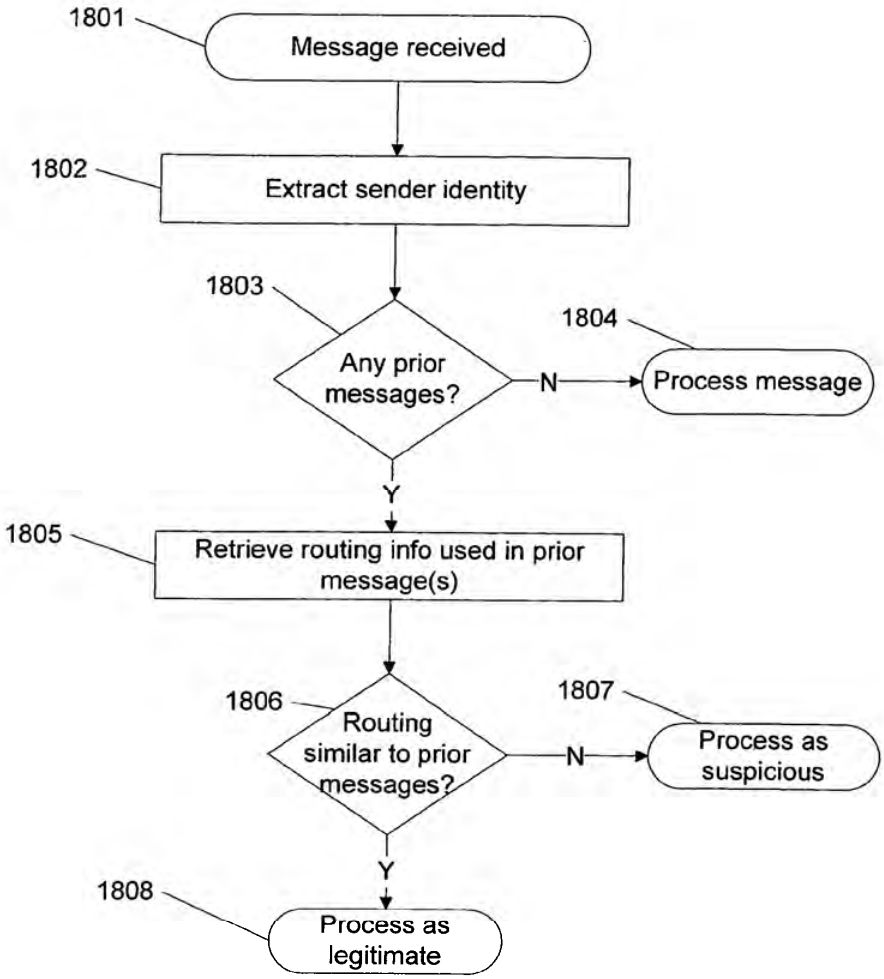

1801
Message received
1802
Extract sender identity
1803
Any prior messages?
N
1804
Process message
Y
1805
Retrieve routing info used in prior message(s)
1806
Routing similar to prior messages?
N
1807
Process as suspicious
Y
1808
Process as legitimate

Figure 18

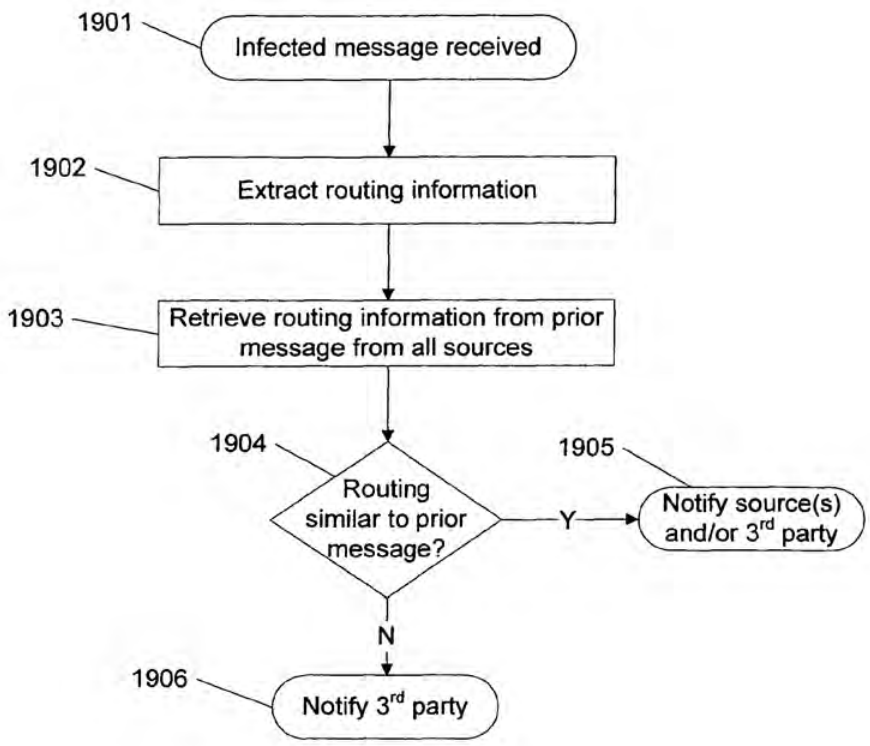

1901
Infected message received
1902
Extract routing information
1903
Retrieve routing information from prior message from all sources
1904
Routing similar to prior message?
Y
1905
Notify source(s) and/or 3rd party
N
1906
Notify 3rd party

Figure 19A

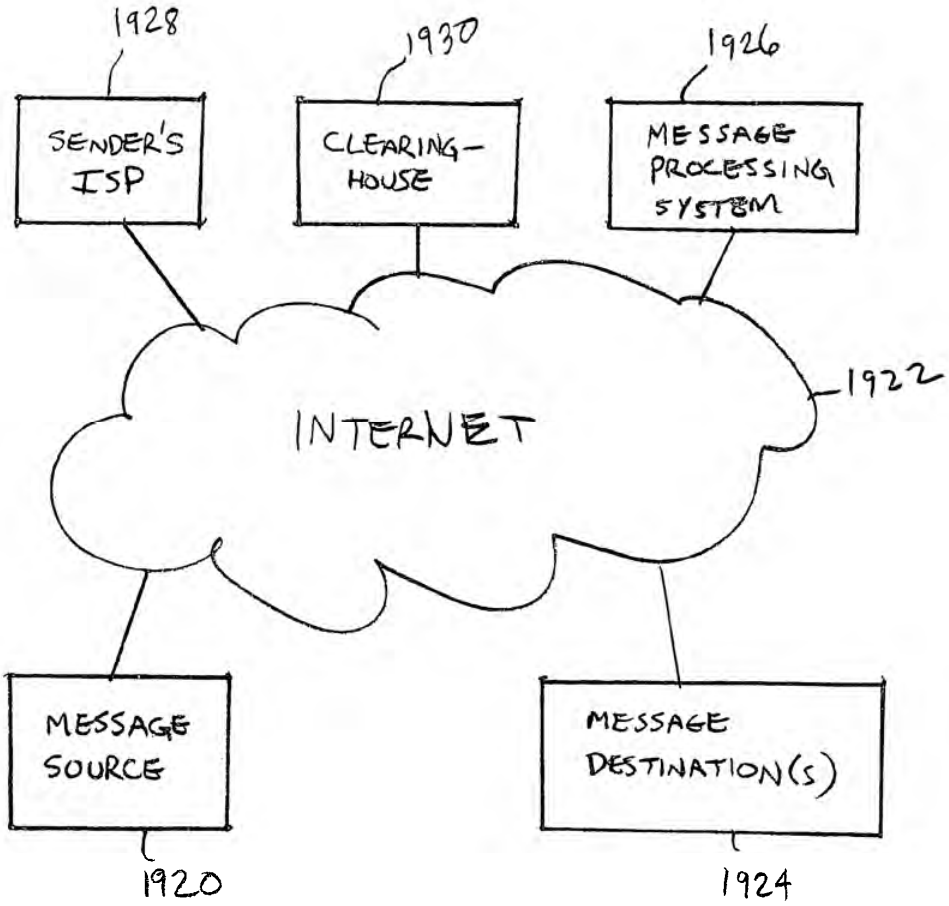

1928
SENDER'S ISP
1930
CLEARING-HOUSE
1926
MESSAGE PROCESSING SYSTEM
INTERNET
1922
MESSAGE SOURCE
1920
MESSAGE DESTINATION(S)
1924

FIG. 19B

CONTAGION ISOLATION AND INOCULATION

CROSS REFERENCE TO OTHER APPLICATIONS

This application claims priority to U.S. Provisional Patent Application No. 60/613,909 entitled CONTAGION ISOLATION AND INOCULATION filed Sep. 27, 2004, which is incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

Laptop and wireless computers and other mobile systems pose a threat to elements comprising and/or connected to a network service provider, enterprise, or other protected networks to which they reconnect after a period of connection to one or more networks and/or systems that are not part of the service provider, enterprise, or other protected network. By roaming to unknown domains, such as the Internet, and/or connecting to such domains through public, wireless, and/or otherwise less secure access nodes, such mobile systems may become infected by computer viruses, worms, backdoors, and/or countless other threats and/or exploits and/or have unauthorized software installed; have software installed on the mobile system by an operator of the protected network for the protection of the mobile system and/or the protected network removed or altered without authorization; and/or have configurations, settings, security data, and/or other data added, removed, and/or changed in unauthorized ways and/or by unauthorized person. Similarly, a stationary system may become infected, e.g., due to receipt and execution of malicious code via a network or other communication and/or a diskette and/or other removable media. Upon connecting to a protected network, a system may infect or otherwise harm resources associated with the protected network before measures can be taken to detect and prevent the spread of such infections or harm. Therefore, there is a need for a reliable way to ensure that a system does not infect or otherwise harm other network resources when connected to a protected network.

BRIEF DESCRIPTION OF THE DRAWINGS

Various embodiments of the invention are disclosed in the following detailed description and the accompanying drawings.

FIG. **1** is a flow diagram of a method for reducing an anti-contagion update interval during periods of suspicious network activity according to some embodiments.

FIG. **2**A is a flow diagram of a method for cooperatively scheduling an anti-contagion update interval according to some embodiments.

FIG. **2**B is a block diagram of a network environment in which anti-contagion updates are distributed according to some embodiments.

FIG. **3**A is a flow diagram of a method for contacting an anti-contagion source and optionally getting an update according to some embodiments.

FIGS. **3**B-**3**D illustrate distributed update distribution as implemented according to some embodiments.

FIG. **4**A is a flow diagram of a method for acquiring an update according to some embodiments.

FIG. **4**B is a block diagram of a client system configured to obtain an update according to some embodiments.

FIG. **4**C is a block diagram of a system used to distribute an update according to some embodiments.

FIG. **5** is a flow diagram of a method for distributing updates according to some embodiments.

FIG. **6** is a flow diagram of a method for establishing a redistribution policy according to some embodiments.

FIG. **7** is a flow diagram of a method for redistributing content such as updates according to some embodiments.

FIG. **8** is a flow diagram of a method for using peers or other indirect data sources to acquire content such as updates according to some embodiments.

FIG. **9** is a flow diagram of a method for redistributing content such as updates according to some embodiments.

FIG. **10**A is a flow diagram of a method for identifying and remedying problems according to some embodiments.

FIG. **10**B is a block diagram illustrating a network environment in which infected hosts and/or networks are quarantined according to some embodiments.

FIG. **11** is a flow diagram of a method for identifying vulnerabilities according to some embodiments.

FIG. **12** is a flow diagram of a method for identifying an infestation according to some embodiments.

FIG. **13** is a flow diagram of a method for monitoring one or more computers for infestation according to some embodiments.

FIG. **14** is a flow diagram of a method for quarantining a computer according to some embodiments.

FIG. **15** is a flow diagram of a method for quarantining a computer according to some embodiments.

FIG. **16** is a flow diagram of a method for a quarantine server to respond to requests according to some embodiments.

FIG. **17** is a flow diagram of a method for scanning backdoors and inoculating according to some embodiments.

FIG. **18** is a flow diagram of a method for categorizing messages according to some embodiments.

FIG. **19**A is a flow diagram of a method for notifying possible sources of infection according to some embodiments.

FIG. **19**B is a block diagram illustrating a network environment in which suspicious messages are identified and/or one or more notifications are sent when a malicious message is detected.

DETAILED DESCRIPTION

The invention can be implemented in numerous ways, including as a process, an apparatus, a system, a composition of matter, a computer readable medium such as a computer readable storage medium or a computer network wherein program instructions are sent over optical or electronic communication links. In this specification, these implementations, or any other form that the invention may take, may be referred to as techniques. A component such as a processor or a memory described as being configured to perform a task includes both a general component that is temporarily configured to perform the task at a given time and a specific component that is manufactured to perform the task. In general, the order of the steps of disclosed processes may be altered within the scope of the invention.

A detailed description of one or more embodiments of the invention is provided below along with accompanying figures that illustrate the principles of the invention. The invention is described in connection with such embodiments, but the invention is not limited to any embodiment. The scope of the invention is limited only by the claims and the invention encompasses numerous alternatives, modifications and equivalents. Numerous specific details are set forth in the following description in order to provide a thorough under-

standing of the invention. These details are provided for the purpose of example and the invention may be practiced according to the claims without some or all of these specific details. For the purpose of clarity, technical material that is known in the technical fields related to the invention has not been described in detail so that the invention is not unnecessarily obscured.

Contagion isolation and inoculation is disclosed. In some embodiments, a request is received from a host, e.g., via a network interface, to connect to a protected network. It is determined whether the host is required to be quarantined. If the host is required to be quarantined, the host is provided only limited access to the protected network. In some embodiments, a quarantined host is permitted to access the protected network only as required to remedy a condition that caused the quarantine to be imposed, such as to download a software patch, update, or definition; install, remove, and/or configure software and/or settings as required by a policy; and/or to have a scan or other diagnostic and/or remedial operation performed. In various embodiments attempts to communicate with hosts not involved in remediation are redirected to a quarantine system, such as a server, that provides information, notices, updates, and/or instructions to the user.

Contagion refers herein to software programs and/or code, data values, configurations, and settings, that expose a computer or other system to security threats, such as unauthorized access and/or use of the system and/or data stored thereon; corruption or loss of data; and/or damage, loss, and/or unauthorized reconfiguration of system software and/or hardware components. Examples of contagion include software programs and/or code designed to perform undesirable and/or unauthorized actions from the perspective of the (authorized) user of a computer, and at the same time perform desired actions from the perspective of the creator or controller of such contagion. Examples of contagion include computer worms, viruses, and Trojan horses. Anti-contagion software refers herein to software that prevents, impedes or remediates contagion, such as Norton Antivirus from Symantec, or VirusScan from McAfee, which identifies and/or removes contagion from a user's computer. Another example of anti-contagion software is software that repairs a security vulnerability through which contagion may spread. Many anti-contagion methods rely on contagion descriptions, such as virus signatures, to identify the contagion, and must be updated regularly to be effective.

FIG. **1** is a flow diagram of a method for reducing an anti-contagion update interval during periods of suspicious network activity according to some embodiments. An update interval refers herein to a period of time between acquisition of updates, such as security updates, virus signature file updates, or virus scanner updates. In this example, network traffic statistics may be monitored (**101**). An example of traffic statistics is latency experienced by email, such as average latency, or median latency. Another example of traffic statistics is the rate of unacceptable attempted TCP/IP traffic, such as attempts to connect to blocked or restricted ports, or insufficiently authenticated (or authorized) connections to ports, or connections that provided invalid or illegal content such as extraordinarily long strings, or specific content detected to be malicious. Recent activity may be evaluated (**102**) to see if it is anomalous. Examples of anomalous activity include median email latency in mail queues beyond some threshold, for example above a 4 hour median queuing delay, and rates of unacceptable attempted TCP/IP traffic beyond some threshold, for example greater than two attempts per minutes over a one hour period. Another example of a threshold for anomalous activity is a comparison with recent historical statistics, such as unacceptable attempted TCP/IP traffic at a rate greater than ten times the expected rate over a 30 minute period. The expected rate may for example be an average over a recent historic period, such as an average rate over the last week of connectivity. If activity is not deemed anomalous (**103**), then in this example, processing continues with no change (**104**) in the next update interval. If the activity is deemed anomalous (**103**), then in this example, the next update interval is reduced, for example reduced to some proportion of the planned update interval, for example reduced to 25% of the previous or default update interval. Another example of reducing the next update interval is to reduce the currently remaining time until an update, for example reducing by a proportion, such as 50%, the remaining time until an update is requested.

FIG. **2**A is a flow diagram of a method for cooperatively scheduling an anti-contagion update interval according to some embodiments. In this example, the planned time for contacting an anti-contagion source may arrive (**201**). An old version identifier may be retrieved (**202**), such as a checksum or hash for a file, or a version number for software stored in a file system or registry, or a version number for an anti-virus signature file stored in a file system or registry. An update may be requested from an anti-contagion source (**203**), for example as is discussed in conjunction with FIG. **3**A, which may include updating or concluding that an update is not needed. A time for next contacting an anti-contagion source is decided upon (**204**). In some embodiments, the scheduled time may be made sooner when the source of the update, such as a vendor of software to which the update pertains, anticipates the impending release of a fully tested update, such as scheduling a next contact time in a few hours, or a few days. If the current computer's version of anti-contagion software contains resolutions to all known problems and otherwise anomalous traffic patterns then, in this example, the update interval may be made longer, such as the full default interval for a computer, for example one week. In some embodiments, the selection of the next update interval may be dependent on the specific product. For example, a "pro version" may have a short default interval, such as one day, and a "home version" may have a relatively long default interval, such as two weeks or one month.

FIG. **2**B is a block diagram of a network environment in which anti-contagion updates are distributed according to some embodiments. A plurality of clients **1** to "n", represented in FIG. **2**B by clients **220**, **222**, and **224**, are connected to the Internet **226**. An update server **228** also is connected to the Internet **226**. In various embodiments, one or more of the clients **220**-**224** is configured to request and obtain updates, if any, from update server **228** via the Internet, e.g., as described above in connection with FIG. **2**A.

FIG. **3**A is a flow diagram of a method for contacting an anti-contagion source and optionally getting an update according to some embodiments. In this example, an old version identifier may be retrieved (**301**), for example as was discussed in **202** of FIG. **2**A. Contact may be made with an anti-contagion source (**302**) via a network such as the Internet. The old version identifier may be provided, for example using a transport protocol such as TCP/IP, and a data encapsulation format such as XML, or using a higher level protocol and format such as SOAP. A decision may be made as to whether an update is needed (**303**). In some embodiments, this update decision may be made by a server, upon review of the old version identifier. In some embodiments, this update decision may be made by the user's computer. In some embodiments, the update decision may be made by incorporating preferences held on a user's computer, such as a pref-

erence to always have the newest updates, or preferences to avoid updates that have been available for less than a threshold length of time (for example, 2 weeks), or preferences that accept an additional update when the source indicates urgency associated with the update. In some embodiments, the update decision may be made based on considerations such as the size of the potential update, for example by avoiding the overhead of small updates, and only performing an update when a threshold amount of change has taken place, or when a threshold length of time has elapsed since an actual update was made. An example of a threshold amount of change taking place is the addition of a prescribed number of viral signatures, such as 2 new signatures. An example of a threshold length of time is 3 weeks. If the update is not needed (**303**) then the process of contacting the source is complete (**304**) in this example. If an update is needed (**303**) then an update is acquired in this example via direct or indirect channels (**305**), for example as further described in conjunction with FIG. **4**A. After the update has been acquired (**305**), the contact may be complete (**306**).

Distributed update distribution is disclosed. In some embodiments, at least some users receive an update indirectly, e.g., from another user that received it previously, either directly from the original source or indirectly from a source that had itself earlier received it, again either directly or indirectly, and so on. FIGS. **3**B-**3**D illustrate distributed update distribution as implemented according to some embodiments. In the example shown in FIGS. **3**B-**3**D, it is assumed for expository purposes that each host that receives the update will serve as a redistribution node, each recipient has the ability to redistribute at the same rate as the original source, and there is continuous demand for the update from all sources. As shown in FIG. **3**B, at a time t**1** a first direct recipient of the update D**1** (**332**) downloads the update from the original source S (**330**). As shown in FIG. **3**C, at a time t**2** a second direct recipient D**2** (**334**) downloads the update from the original source and an indirect recipient R**11** (**336**) downloads the update from the first direct recipient D**1** (**332**), which in this case acts as a redistribution node. As shown in FIG. **3**D, at a time t**3** a third direct recipient D**3** (**338**) downloads the update from original source S (**330**), a second indirect recipient R**12** (**340**) downloads the update from the first direct recipient D**1** (**332**), a third indirect recipient R**21** (**342**) downloads the update from the second direct recipient D**2**, and a fourth indirect recipient R**111** (**344**) downloads the update from first indirect recipient R**11** (**336**). In this example, in the three download cycles shown in FIGS. **3**B-**3**D on its own the original source would only have been able to download the update to the three direct recipients, but by using previous recipients to redistribute, in the example shown in FIGS. **3**B-**3**D a total of seven recipients have been able to obtain the update. In a generalized case, under the assumptions used in this example in "n" download cycles/periods a total of 2n−1 recipients are able to obtain the update, whereas only "n" recipients, i.e., one per cycle, would otherwise have been able to obtain the update if it were only available directly from the original source. In an even more generalized case, under an assumption that each server may download roughly k copies in a time interval, for example where more than one download is provided by a server at a time, then in "n" download cycles (or periods) a total of roughly (k+1)n−1 recipients are able to obtain the update, where as only k*n recipients, i.e., k per cycle, would otherwise have been able to obtain the update if it were only available directly from the original source.

FIG. **4**A is a flow diagram of a method for acquiring an update according to some embodiments. In this example, an update source is contacted (**401**), such as via an Internet connection, for example via TCP/IP. An offer to serve as a redistribution source is made (**402**). In some embodiments, a redistribution source may be contacted by other computers to acquire an update. An example of an offer to be a redistribution server is described in conjunction with FIG. **6**. In various embodiments, **402** is optional, performed only if a criterion or test is satisfied, or omitted. An update or an update identifier is acquired (**403**). An example of an update identifier is a version number for an update. Another example of an update identifier is a checksum for an update. Another example of an update identifier is a set of checksums (such as MD5, SHA-1, or CRC-32) for a corresponding set of byte ranges that form the update. Another example of an update identifier is a combination two or more identifiers, such as the identifiers enumerated above. If the update was completed (**404**) then the user confirms redistribution sourcing status (**408**) in this example. In some embodiments, confirmation includes asserting that the client is prepared to redistribute copies of the received data to one or more recipients. If the update was not completed (**404**), then a list of peer or other sources is acquired (**405**) in this example. In some embodiments, the list is or may be explicit. In some embodiments, the list is or may be implicit, for example represented by an iterator or accessor, which may retrieve elements on an as needed basis. In some embodiments, implicit lists are dynamic, and calculated or created in part on an as needed or as requested basis. In some embodiments, the list of other sources contain addresses that may be contacted to acquire an update, such as a list of IP addresses and ports, as well as optionally some authentication information to be used when contacting a listed source. In some embodiments, a requesting host that agrees to serve as a redistribution source is provided with a list of sources from which the update is available immediately, or at least sooner than it will be made available to hosts that will not redistribute. One or more attempts may be made to acquire the update from a listed source (**406**), for example as exemplified in FIG. **8**. If the update is completed via a listed source (**407**), then redistribution status is confirmed (**408**) in this example. If the update is not complete (**407**) then a reservation is optionally negotiated and/or acquired (**409**) in this example. In some embodiments, a reservation may include a time and/or a specific source or IP address. In some embodiments, a reservation address may be selected from a collection of source addresses, for example in order to distribute the load across more than one server, or to provide a server that is closer to a recipient. An example of selecting a server close to a recipient is to select a server having relatively lower latency during a download, or relatively lower cost, for example by selecting a source within a recipient's ISP complex or within a recipient's intranet or home network. If a reservation time was acquired, then a wait for that reservation time (**410**) is performed in this example. At the reservation time, a further attempt to acquire an update or update identifier from the source (**403**) is made, and the flow proceeds as described earlier. This flow may continue until an acquisition is complete (**408**), as also described earlier. In various embodiments, one or more of confirming redistribution sourcing status (**408**), attempting to obtain the update from a redistribution source (**405**-**407**), and/or acquiring and/or waiting for a reservation time (**409**-**410**) is/are optional, conditional, or omitted.

FIG. **4**B is a block diagram of a client system configured to obtain an update according to some embodiments. In some embodiments, the client system shown in FIG. **4**B is used to implement one or more of clients **220**-**224** of FIG. **2**B. The client system **420** includes a communication interface **422**,

such as a network interface card, configured to send/receive communications via a network, such as the Internet. The client system **420** further includes an update agent and/or browser **424**. In some embodiments, the update agent/browser **424** comprises browser software used to navigate the World Wide Web, e.g., to send/receive data to/from a website from which an update is available for download. In some embodiments, update agent/browser **424** includes an agent associated with anti-contagion software installed on the client system **420**. In various embodiments, the update agent/browser **424** comprises one or more components and is implemented as one or more software programs running on a processor and/or one or more hardware components. A settings data store **426** stores one or more settings and/or other data associated with the software with which an update and/or software to which the update does or may pertain, such as periodicity or next scheduled (e.g., reservation) time to check for updates, a list of update identifiers, locators, and/or redistribution sources from which an update is or may be available, policies for whether and/or under what terms, conditions, circumstances, etc. the client system **420** may be used redistribute an update, a last update time, an old version identifier (see, e.g., **202** of FIG. **2**A), etc.

FIG. **4**C is a block diagram of a system used to distribute an update according to some embodiments. In some embodiments, the system of FIG. **4**C is used to provide an update server, such as update server **228** of FIG. **2**B. In this example, the distribution system **440** includes a communication interface **442** configured to send/receive data via a network, such as the Internet. An update distribution engine **444** is configured to receive and service requests for updates. For updates provided directly to a request source, the update distribution engine **444** obtains the update from an update storage location **446** and provides the update to the request source via the communication interface **442**. In some embodiments, if data included in the request indicates the source of the request is or may be willing to serve as a redistribution source, the update distribution engine **444** hands the request off to a redistribution coordinator **448**, which negotiates with the request source, e.g., based on policies, settings, rules, etc. stored in a policies and lists data store **450**, through communications sent/received via communication interface **442**, to determine if the requesting source will serve as a redistribution source and, if so, to determine the terms and conditions of such service and data, such as an address, locator, or identifier, that can be provided to other requesting systems and/or users to enable them to attempt to obtain the update from the redistribution source. In some embodiments, if update distribution engine **444** determines that it is not able to provide the update to the request source directly (e.g., due to capacity limits or because a policy applicable to the request and/or the request source indicates the request source should be directed to a redistribution source, e.g., because the request source has a less expensive version of the software or is not willing to serve as a redistribution source), the update distribution engine **444** hands the request off to redistribution coordinator **448**, which provides to the request source an update identifier and a list of one or more redistribution sources from which the update is or may be available to the request source.

FIG. **5** is a flow diagram of a method for distributing updates according to some embodiments. In this example, a source prepares to supply updates to recipients (**501**), for example by starting up a server, for example an HTTP or FTP server, or a multicast server, such as an RTP multicast server. The server waits for an event (**502**), such as a contact on a predefined port, or a notification that the server should start a multicast. If a multicast-time event appears (**503**) then an update is multicast (**504**) in this example. In some embodiments, a multicast time event may be generated internally or externally to the server to start a multicast. In some embodiments, a multicast-time event may be automatically generated at predetermined times, such as hourly, weekly, or daily, or on individually specified dates and times. In some embodiments, a multicast-time event may include a specific identifier for the multicast. If an event of the form of an update request arrives (**503**) then a test for authenticity is optionally performed (**505**) in this example. An example of determining whether a request is authentic is to check a credential. Examples of credentials include a provided shared secret, such as a secret used by all requestors to this server, or a different secret used by each requestor such as a user serial number, or a cryptographically signed authentication, such as credentials or a certificate issued to users that had paid for an update service, for example, by inclusion in a purchased software package. In some embodiments, all requests are assumed to be authentic without the use of any authenticator. If the request was not authentic (**505**) then, in this example, the server continues to wait for an event (**502**). If the request was authentic (**505**) then the server may determine a necessary update (**506**) in this example. In some embodiments, determination of an update may be made based on the requestor's state, which in some embodiments may be indicated in a received request for update. A redistribution contract may be negotiated (**507**). In some embodiments, some requestors may agree to redistribute updates to other requestors. In some embodiments, some requestors may decline to redistribute updates. In some embodiments, a source may decline redistribution proposals from a requestor. An example of a redistribution contract is an agreement not to redistribute. Another example of a redistribution contract is an agreement to redistribute a fixed number of copies, such as five copies. Another example of a redistribution contract is an agreement to redistribute for a fixed period of time, such as one day. Another example of a redistribution contract is an agreement to redistribute within a specified geographic region within a fixed period of time. Another example of a redistribution contract is an agreement to redistribute indefinitely, for example until a replacement update is provided. A redistribution contract may be made with terms that are combinations of the above mentioned exemplary term limitations. An update may be provided directly or indirectly (**508**). A direct update may for example be provided by allowing an FTP download, or by encapsulating the transmission in an HTTP transfer, for example using XML and/or an appropriate coding, such as uuencode. An indirect update may be provided by directing the requestor to an alternate source, such as a redistribution server, or a server at a different address, or a server at a different time, such as a future time, or a combination of two or more such specifications. In some embodiments, a decision to provide direct vs. indirect updates may be based in part on the redistribution contract, for example a requestor that is willing to redistribute may be updated sooner or more directly. An example of a redistribution contract is a promise by a user to provide redistribution in exchange for purchase price concession, such as a reduced initial purchase price, or a quantity discount price. In some embodiments, a decision to provide direct vs. indirect updates may be based in part on previous agreements, such as prepayment for expedited service, such as may be included in a more expensive version of a product that utilizes the updates. The server may wait for an event (**502**). In some embodiments, events may be processed in parallel with previous events, or may be sequentially pro-

cessed, or some combination, for example up to a threshold number of events in parallel, such as a maximum of 32 events in parallel.

FIG. **6** is a flow diagram of a method for establishing a redistribution policy according to some embodiments. In this example, a computer may prepare to offer to be a redistribution source (**601**). In some embodiments, preparations may include retrieving preferences, such as the number of redistributions that may be done, or the duration of time that the computer is willing to act as a redistribution source. In some embodiments, preferences may be defaults, or may be configured by a user and stored persistently, such as in a registry or file system. A contact address may be proposed to the source and a policy may be negotiated (**602**). A contact address may for example be a globally routable IP address and a port, or a local IP address (reachable from a LAN behind a firewall) and port that can be accessed by other computers on the LAN, such as computers that shared a common NAT address in their contact with the original source. In some embodiments, negotiation may include proposing a policy. In some embodiments, negotiations may involve a counter proposal in response to a proposal. An example of a counter proposal criterion is to require a minimum duration for redistribution. If terms were not acceptable (**603**) then, in this example, the computer does nothing (**604**). If the terms were acceptable (**603**) then, in this example, the computer prepares to service redistribution requests (**605**), for example by establishing a server to respond at the agreed contact address, reserving storage to hold the content for redistribution, etc. In some embodiments, redistribution policy may be a broader contractual matter, where an update is provided more quickly or more directly in exchange for a promise to redistribute. In some embodiments, an auditing mechanism may be used to measure compliance and penalize contract breakers, such as failure to accept connections for redistribution, where penalties may for example include refusal to accept future similar contracts. As another example, a lower purchase or subscription price may be offered for operators of a software product, such as antivirus software, who agree to act as redistribution points. In some embodiments such purchase agreements may be configured into the software, for example by establishing agreements to act as a redistribution point for updates as defaults that a user cannot modify.

FIG. **7** is a flow diagram of a method for redistributing content such as updates according to some embodiments. In this example, a redistribution computer is contacted and confirms its sourcing status (**701**), for example by accepting a TCP/IP connection on a predefined port, and processing a request for content. The redistribution computer may optionally contact another computer, for example a more original source, and affirm its redistribution sourcing status (**702**). A more original source may be a source that has provided content in the past to the redistribution computer, directly or indirectly. In some embodiments, contacting a more original source demonstrates that the redistribution computer is performing redistribution. In some embodiments, contacting a more original source allows the redistribution computer to further authenticate the computer that requested the content, for example by getting confirmation from a more original source that a specific computer was authorized to request the content. An authentication to be used to authenticate requests is negotiated (**703**). In some embodiments, authentication includes providing a shared secret to the redistribution computer. In some embodiments, authentication includes presenting an authorization cryptographically signed by a more original source, such as a certificate. In some embodiments, all requests are presumed authentic. If the request is not authentic (**704**), then nothing further is done (**705**) in this example. If the request is authentic (**704**), then the requested update is supplied (**706**) in this example. The updates may be supplied for example via a transport protocol TCP/IP, or a higher level protocol such as FTP, HTTP, or HTTPS, etc.

FIG. **8** is a flow diagram of a method for using peers or other indirect data sources to acquire content such as updates according to some embodiments. In this example, a list of one or more peers or other indirect sources for acquiring updates is obtained (**801**), for example as was discussed in conjunction with **405** of FIG. **4**A. In one example, an indirect source may be a peer, such as another computer that has downloaded the content. In another example, an indirect source may be a computer configured to redistribute content without other use of the content, such as an ISP server established to redistribute content to subscribers. In another example, an indirect source may be a server created or authorized by the original source to perform additional distributions. In some embodiments, an obtained list of indirect sources may be implicit, for example by referring to a list using an iterator or accessor construct that can itemize entries in the list when requested, for example using a query to a service when an additional item in the list is required, which may for example dynamically complete the list on an as requested basis. A next address and (optional) authenticator in the list of sources is retrieved (**802**). Any order may be used. In some embodiments, an order of processing may correspond to the original list order. In some embodiments, an order of processing may be such that the earlier processed addresses would be expected to have lower latency and/or higher bandwidth, for example by first processing a local address on a LAN. In some embodiments, elements of the list are or may be processed in parallel, and portions (such as byte ranges) of the content may be acquired from different addresses in the list. An address on the list may be contacted to request content (**803**), which may in some embodiments be performed as described in conjunction with FIG. **7**. In one example, a source address may consist of an identifier for a broadcast with optional time(s) for a broadcast, and contact with the source may include receiving the broadcast. In another example, contact with the source may include making a connection such as a TCP/IP connection with the source. If the acquisition is complete (**804**), then the processing is complete (**805**) in this example. If the acquisition was not completed (**804**) then additional sources in the list are considered in this example. If there are more sources in the list (**806**) then, in this example, a next source is retrieved along with an optional authenticator (**802**) and the processing continues. If there are no more sources (**806**) then the attempted acquisition from source(s) is unsuccessful (**807**) in this example.

FIG. **9** is a flow diagram of a method for redistributing content such as updates according to some embodiments. In this example, a computer prepares to supply content (or updates) to a requestor/recipient (**901**), for example by starting up one or more servers. Examples of a server include an HTTP server, an FTP server, and a multicast server. A server waits for an event (**902**), such as a contact on a predefined port, or a notification that the server should be terminated, or a notification that a time for a multicast has arrived. If a service expiration event appears (**904**) then the supply service is terminated (**905**) in this example. A service expiration event may be automatically generated, for example after a predetermined duration of service, or after a predetermined number of copies of an update have been redistributed. If a request event arrives (**904**), then a test for authenticity is performed (**906**) in this example. Authenticity may be established in the request, for example by receiving and validating, or perform-

ing a mutual proof of possession of, a shared secret, such as a secret used by all requestors to this server, or a different secret used by each requestor of this service and held in a list of single-use secrets by this server, or by providing a cryptographically signed authentication. In some embodiments, all requests are considered authentic without the use of any authenticator. If the request was not authentic (**906**) then, in this example, the server continues to wait for an event (**902**) as described earlier. If the request was authentic (**906**) then, in this example, the server transmits a copy of the content to the requestor (**907**), for example by allowing an FTP download, or by encapsulating the transmission in an HTTP transfer, for example using XML and/or an appropriate coding, such as uuencode.

FIG. **10**A is a flow diagram of a method for identifying and remedying problems according to some embodiments. A problem may be identified as some event or status that may cause direct or potential harm. Exemplary problems include vulnerability or infestation. In this example, a problem such as a vulnerability or infestation is detected (**1001**). Infestation herein refers to the current presence and/or execution of contagion. In some embodiments, detection may be made by a router, firewall, or other network component, separately or in concert, using techniques such as those illustrated by FIGS. **11**, **12**, and **13**. In some embodiments, detection may be made by a computing device, for example a computing device operated by an ISP. After a detection of a problem, a computer or network containing the problem is quarantined (**1002**). In some embodiments, quarantining includes preventing unauthorized communications from being sent from the quarantined computer, such as efforts to infect, probe, or harass other computers, or efforts to communicate data without authorization, including for example file contents and keyboard entry logs. In some embodiments, no communications are authorized. In some embodiments, quarantining includes preventing undesired communications from reaching the quarantined computer, such as control messages or infectious messages. Examples of quarantining activities are illustrated in FIGS. **14** and **15**. In this example, the quarantined computer is provided access to remediation services (**1003**). In some embodiments, access to remediation includes or may include allowing connections to servers providing security patches or advisories, for example allowing connections to a vendor's server, or providing virus and worm disinfecting tools, such as focused removal tools, or data updates for previously installed repair tools. In some embodiments, access to remediation includes allowing direct access to vendor sites known to provide remediation assistance, such as to Microsoft's Windows Update service, where security patches may be obtained. In some embodiments, access to vendor sites is restricted by the quarantine to certain classes of traffic, such as HTTP requests, certain port access, such as port **80** or port **443**, certain content requests, such as specific URLs or specific file downloads. In some embodiments, remediation content may be substituted for other requested content, for example as discussed in conjunction with FIG. **14**.

FIG. **10**B is a block diagram illustrating a network environment in which infected hosts and/or networks are quarantined according to some embodiments. A plurality of hosts **1** to "m", represented in FIG. **10**B by hosts **1020**, **1022**, and **1024**, connect via a router **1026** and a gateway **1028** to the Internet **1030**. In various embodiments the router **1026** is omitted or included in the gateway **1028**. In various embodiments, gateway **1028** comprises a gateway, router, firewall, or other device configured to provide and control access between a protected network and the Internet and/or another public or private network. In various embodiments, the process of FIG. **10**A is implemented, in the event of quarantine of one or more of hosts **1020**-**1024**, for example, on the affected (quarantined) host, router **1026**, and/or gateway **1028**. In some embodiments, a quarantined host (or a host associated with a quarantined network or sub-network) is permitted to access a remediation server **1032**, e.g., to download a patch, more current threat definition, etc. In some embodiments, requests to connection to a host other than the remediation server **1032** are redirected to a quarantine server **1034** configured to provide a notice and/or other information and/or instructions to a user of the quarantined host. In various alternative embodiments the quarantine server **1034** is omitted or resides on the router **1026** and/or gateway **1028**.

FIG. **11** is a flow diagram of a method for identifying vulnerabilities according to some embodiments. In this example, preparation is made to scan for one or more vulnerabilities (**1101**), for example by retrieving one or more IP addresses of computers that need to be scanned. For example, an IP address may be provided by a DHCP sever hosted in an ISP network complex, to identify current subscriber IP addresses. As another example, a subnet range may be scanned, for example by taking an address of the form A.B.C.X, and using some or all values of X in the range 0 to 255 to enumerate distinct IP addresses. A scan of one or more IP addresses may optionally be performed to detect an open port (**1102**). An example of scanning an IP address is scanning to detect open TCP/IP ports (**1102**). An open port may be identified by attempting to form a connection, such as a TCP/IP connection to that port. In some embodiments, only open ports on a given computer will provide acknowledgment messages needed to establish a TCP/IP connections, which for example causes a success code to be returned by Unix "connect( )" and a closed port will provide no response to the attempt to connect, which for example would cause an error code to be returned by the Unix function call "connect( )". If no open ports are detected (**1103**), then the scanning is complete (**1106**) in this example. If open ports are detected (**1103**) then those ports are tested for vulnerabilities (**1104**) in this example. In some embodiments, an open port is assumed (**1103**) and a test for vulnerability is performed or attempted (**1104**), for example by communicating a vulnerability test without independently testing to see if one or more requisite ports are open. In some embodiments, a vulnerability test of a port may include sending data that is known to create a discernable response when a vulnerability is present, such as sending a long string known to induce a buffer overflow with a consequent errant response, where the errant response may for example include a characteristic communication, premature termination of the connection, or other erroneous response(s). In some embodiments, a test for a vulnerability is an attempt to exercise a service, such as a sendmail or an RPC service, in search of a misconfiguration of that service that creates a vulnerability. One example of a misconfiguration of a service is provision of any such service. Another example of a misconfigured service is one where a password should be required for security purposes, but no password is required by the misconfiguration. Another example of a misconfiguration is the use of an easily guessable password, such as the default installation password, when resulting access creates a security vulnerability. Another example of a misconfigured service is support for a service provided by an unauthorized agent, such as a worm or virus. If no vulnerabilities are detected (**1105**) then scanning is complete (**1106**) in this example. If vulnerabilities are detected (**1105**) then an associated IP address is flagged as vulnerable (**1107**) in this example. In the case of a subscriber to an ISP, an IP address may be translated to a particular subscriber identity, for

example by checking in DHCP or static IP allocation tables for the subscriber's identity, or for example by extracting a MAC address from traffic associated with the IP address, and such identity may be appropriately flagged or recorded. In some embodiments, the record of the vulnerability may additionally include the port, and the details of the vulnerability that was detected, such as which buffer overrun was vulnerable, or which service was misconfigured to create a vulnerability.

FIG. **12** is a flow diagram of a method for identifying an infestation according to some embodiments. In this example, a device near a computer, such as a firewall on a LAN or an ISP's router, is used in some embodiments to monitor a computer for infestation (**1201**). Network traffic is monitored for anomalous traffic volume, port, or IP destination (**1202**), for example by detecting large volumes of outbound traffic, such as email traffic or instant messaging traffic. Another example of anomalous traffic is a TCP/IP connection to a server that is known to control a collection of infested machines, in a way that suggests the connecting machine is controlled by the server, such as connecting and transmitting an assertion that it is controlled or available to comply. Another example of anomalous traffic is a connection to a computer that is a current target of a denial of service attack, with a pattern that is consistent with the attack, such as a connection to a targeted port, or the use of a falsified return IP address. In some embodiments, a falsified return address may be detected by a router, such as an ISP router, that is aware that the return address is not on the sourcing subnet, or not associated with the MAC address that has supplied the corresponding packets. If an anomaly is detected (**1203**) then an apparent infestation is noted (**1204**) in this example. In some embodiments, a response to infestation may include quarantining of the computer or network containing the computer, for example as discussed in conjunction with FIGS. **14**, **15**, and **16**. In some embodiments, a response to infestation may include an attempt to remediate the problem, for example by inoculation, such as discussed in conjunction with FIG. **17**. If no anomaly is detected (**1203**) then monitoring is complete (**1205**) in this example. In some embodiments, monitoring for anomalous traffic may be continuous. In some embodiments, monitoring for anomalous traffic may be periodic. For example, periodic monitoring may consist of monitoring some number of computers, such as 1 computer, for some period of time, such as 2 minutes, and then not monitoring those computers again for another period of time, such as 1 hour. As another example, periodic monitoring may be randomized, for example by changing the duration of monitoring, pattern of selecting computers to monitor, and/or the unmonitored duration, to be difficult or impossible to predict, for example by being driven by a pseudo random number generator, subject to restrictions of available monitoring computers.

FIG. **13** is a flow diagram of a method for monitoring one or more computers for infestation according to some embodiments. In this example monitoring of a computer for infestation begins (**1301**), for example by retrieving a list of one or more addresses of computers, such as addresses of participating subscribers. In some embodiments, this list may be a list of addresses for some or all subscribers for a given ISP. In some embodiments, the list may be a list of IP addresses issued by a DHCP server, such as on a LAN, or a list of IP addresses observed by a router, such as a router on a LAN, or a complete list of IP addresses in a given subnet. A computer associated with an address identified in a list is queried for a cleanliness assertion (**1302**), for example by contacting a trusted computing base within a computer, and requesting an authenticated infestation scan by trusted software. An example of a trusted computing base within a computer is the Paladium security initiative under development by Microsoft and supported by Intel and American Micro Devices. Another example of a trusted computing base is described in various TCG specifications, such as the TCG Architecture Overview, published by the Trusted Computing Group. Trusted code bases may for example execute antivirus scans of the remainder of the computer, including untrusted portions of the disk and/or operating system. In some embodiments, trusted code bases may digitally sign assertions about the cleanliness (e.g. infestation status) and/or state of their computers. In some embodiments, the query for cleanliness (**1302**) may be responded to by anti-contagion software, such as antivirus software, with assertions about the currency of a scan, such as the last time a scan was performed, or a version associated with a current anti-contagion software or definition file in use, wherein a sufficiently updated software and/or scan may act as a cleanliness assertion. In some embodiments, an operating system may respond with information associated with its patch level, wherein a sufficiently recent patch level may be interpreted as an assertion of cleanliness. If a computer asserts it is clean (**1303**) then monitoring is complete (**1305**) in this example. If a cleanliness assertion is not provided (**1303**) then an infestation or vulnerability is presumed (**1304**) in this example.

FIG. **14** is a flow diagram of a method for quarantining a computer according to some embodiments. In this example, a decision has been made to quarantine a computer (**1401**), such as an ISP deciding to quarantine a subscriber, or a firewall or router for a LAN deciding to quarantine a local computer. In some embodiments, a decision to quarantine may be made based on apparent infestations or apparent vulnerabilities, for example as illustrated in conjunction with FIG. **11**, **12**, or **13**. In some embodiments a decision to quarantine may be based on the accumulation of evidence above a threshold, such as a threshold of one such piece of evidence, or a threshold of two such pieces of evidence. In some embodiments, a decision to quarantine may be made based on configuration rules, for example all new computers must be quarantined until they have contacted their OS vendor and been updated, or by user specification, such as a user requesting a quarantine for a specific computer. In some embodiments, requests for a quarantine may be made on an individual computer via settings, such as control panel in Microsoft Windows, or through a user interface to a firewall or router, for example by specifying an IP address, name, or MAC address to quarantine, or a list of addresses that are not to be quarantined (where other addresses may be quarantined). During a quarantine period outbound traffic is detected (**1402**), for example by a device such as a router, or by a software network stack on the quarantined computer, or by an upstream device, such as a device at an ISP. Outbound traffic is tested in this example to see if it is associated with a remediation request (**1403**), such as contact with an appropriate anti-contagion software distributor, or contact with a security patch update provider, such as Microsoft Windows Update, or contact with a verified remediation site, such as an internal ISP site. If outbound traffic is deemed to be associated with a remediation request (**1403**), then the traffic is routed or forwarded to its destination (**1404**) in this example. If outbound traffic is not associated with a remediation request (**1403**), then, in this example, it is tested to see if it is an approved service request (**1405**), such as a request for a web page. If the request was for an approved service (**1405**) then, in this example, the request is re-routed to a special quarantine server (**1407**), such as an ISP web server that

provides information and links to assist in remediation. Re-routing of request traffic may be accomplished in numerous ways, for example by providing an alternate DNS service that consistently returns a fixed IP address of a quarantine server in response to all DNS requests. Another example of re-routing a web page request is to respond with a redirect message, which would direct a browser on a quarantined computer to contact a quarantine server that provides remediation information. If the request was not for an approved service (**1406**), then the outbound traffic is discarded or responded to with in indication that the desired traffic destination is unreachable (**1406**) in this example.

FIG. **15** is a flow diagram of a method for quarantining a computer according to some embodiments. In this example, a decision has been made to quarantine a computer (**1501**), for example as described in conjunction with **1401** of FIG. **14**. Inbound traffic originally destined for a quarantined computer is detected (**1502**), for example by a network device such as by a router. An example of detecting inbound traffic destined for a quarantined computer is to have a list of quarantined IP addresses, and filter or provide additional processing on traffic destined for an address on the list. Detected inbound traffic is tested in this example to see if it is remediation related (**1503**), such as a response to a remediation request. Direct responses to remediation requests may for example be identified using stateful packet inspection firewall techniques known to those skilled in the art. In various embodiments, at least certain types of traffic from approved sites such as anti-contagion vendors, security update providers, and/or internal ISP sites are deemed remediation related. If traffic is remediation related (**1503**) then it is forwarded to the quarantined computer (**1504**) in this example. If traffic is not remediation related (**1503**) then, in this example, it is discarded or responded to with a message indicating that the destination is unreachable (**1507**).

In some embodiments, a configuration based quarantine during an installation of an operating system, such as Microsoft Windows, may be automatically setup, for example by a default installation setting up restrictions in a firewall installed with the OS or by an installation setting defaults for use when network connectivity is enabled, that establishes a quarantine, for example restricting internet connections, for example allowing internet connections only to a security update site such as Microsoft Windows Update. In some embodiments, settings that may establish a quarantine, such as default settings in an operating system installation, may be altered programmatically to remove the quarantine in response to a remediation site, such as a security update service such as Microsoft Windows Update, asserting that there are no other security updates pending. In some embodiments, configuration settings that may establish a quarantine may be altered programmatically to enable a quarantine in response to the presence of one or more security updates, such as provided by an operating system update service such as Microsoft Windows Update, that are available for installation on a computer.

FIG. **16** is a flow diagram of a method for a quarantine server to respond to requests according to some embodiments. In this example, a quarantine server is started (**1601**), for example by starting a web server, or DNS server, or proxy server, or ftp server, or some combination server. The server waits for a request (**1602**), for example by listening on a port and waiting for a connection. In some embodiments, a device such as a router may forward outbound traffic from a quarantined computer to a quarantine server, for example as illustrated in conjunction with **1407** of FIG. **14**. If a received connection is a web server request (**1603**), such as an HTTP request, then the server responds with a quarantine notification page (**1604**) in this example. An example of a quarantine notification page is a web page that provides notification that the computer is quarantined, and/or provides links to remediation sites appropriate to the quarantine, such as a link to a site that provides anti-contagion software for removing a virus that the quarantined computer is believed to contain. The links on the quarantine notification page may be examples of HTTP addresses that are for use in remediation. Another example of a quarantine notification page is a page that redirects to a quarantine notification page. If the request is not a web server request (**1603**), then it is tested to see if it is a DNS inquiry (**1605**) in this example. A DNS inquiry may for example appear on port **53** using TCP/IP or UDP. If the request is not a DNS inquiry (**1605**) then it may be discarded or responded to as "unreachable" (**1606**) in this example. If the request is a DNS inquiry (**1605**) then, in this example, the inquiry is tested to see if it is a DNS request for a remediation host name (**1608**). A remediation host name may include a host name as appropriate to the reason for a quarantine, such as a host providing anti-contagion software, or a host providing security updates, or a host internal to (or partnered with) an ISP. If the DNS inquiry was for a remediation host name (**1608**) then the request is proxied to an external DNS service provider (**1609**) in this example. A proxy to a DNS service (**1609**) may return a stored or cached response, such as the actual IP address of the host name, or it may contact an external DNS service to relay back the proposed IP address. If the DNS inquiry was not for a remediation host name (**1608**) then an IP address for a quarantine server is provided (**1607**) in this example. In some embodiments, when a redirected IP address is provided, such as that of a quarantine server, the DNS response may include a short time to live, for example 0 seconds. A short time to live ensures that a further DNS query will be sent prior to subsequent communications being sent from the quarantined computer to the host that was the subject of the DNS query, aiding a return to normal routing of traffic once the computer is no longer under quarantine. In some embodiments, the server(s) in this example may continue to wait for a service request (**1602**) in parallel with the processing of the requests as described above, or may wait for a service request again after disposition of the previous request as described above.

FIG. **17** is a flow diagram of a method for scanning backdoors and inoculating according to some embodiments. In this example, preparations are made for scanning a computer for a backdoor inoculation path (**1701**), for example by identifying the computer to be scanned, and enumerating or retrieving a list of one or more backdoors. In one example, an IP address to be scanned may be selected by an ISP from a list of subscribers. In one example, a list of backdoors may include information associated with preloaded software approved by the computer owner for access and inoculation purposes, such as an ISP provided utility that is installed by the subscriber. In some embodiments, an ISP defined backdoor may have an authentication protocol to preclude control by unauthorized parties, such as restrictions on IP address(es) that may contact the backdoor, or shared secrets such as a password required by the sender, or cryptographic authentication such as client authentication on an SSL connection. In some embodiments, a user is requested and/or required to authorize in advance use of an authorized and/or any unauthorized backdoor for purposes of inoculation. As another example, a list of backdoors may include virus or worm installed backdoors, such as those installed by a SubSeven virus or Back Orifice. As another example, backdoors may include exploitable security vulnerabilities, such as a weak-

ness in a web server, or a weakness in an RPC service. A scan may optionally be conducted to detect open TCP/IP backdoor ports (**1702**) similarly to the scan discussed in conjunction with FIG. **11**. If none of the backdoors are open (**1703**), for example there is no response on any of the ports, then the scan is complete (**1706**) in this example. If one or more backdoor ports are open (**1703**) then they are tested for usability (**1704**) in this example. In some embodiments, a backdoor port may be assumed open (**1703**) and may be tested for usability (**1704**) without an independent test to see if the port is open. An example of a usability test is no test at all, with the conclusion that a specific port is usable any time it is open. Another example of a usability test is based on the receipt of confirmation of a functional installation of ISP provided backdoor software, such as a confirmation that may be provided by the installed software. Another example of a usability test involves the use of a less invasive test of vulnerabilities such a buffer overrun exploit that does not have a permanent impact but provides observable validation (such as a dropped connection). If a backdoor is usable (**1705**) then the computer is inoculated via a backdoor (**1707**) in this example. One example of such an inoculation is the remotely controlled download of one or more remedy files into one or more directories, such as an operating system directory, such as the "\windows\system" directory in Microsoft Windows 2000. Another example of an inoculation is the download of an executable file to the computer, followed by the execution of that file. Another example of an inoculation is the download of a data file, such as an anti-contagion signature file, that is interpreted or processed by another executable already on the computer. Another example of an inoculation is the deletion of one or more files from the computer. Another example of an inoculation is the editing of one or more registry entries, or of one or more files, such as deleting, appending or rewriting entries in a registry or file. Another example of inoculation includes a combination of one or more of above techniques, optionally in conjunction with one or more remotely instigated reboot operations. Another example of inoculation includes a combination of one or more of the above techniques, along with an optional dialog box requesting a user's permission to proceed with the inoculation. Another example of inoculation is the use of a dialog box in conjunction with a timer that causes the inoculation to proceed if no user response is given in a prescribed period of time, such as 1 hour.

FIG. **18** is a flow diagram of a method for categorizing messages according to some embodiments. In this example, a message is received (**1801**), such as an email message or instant message. The purported sender's identity is extracted (**1802**), for example by examining the "From:" header line in an email message, or by examining the envelope information conveying an email message, or by examining the sender's ID in an instant message. A check is performed to see if the sender has sent any prior messages (**1803**), for example by scanning an inbox of prior email, or by checking for the sender in a buddy list. If there were no prior messages from the sender (**1803**) then the message is processed (**1804**) in this example. Processing the message may for example include presenting the message to the user, and/or recording routing information for comparison and categorization of future messages. If there were prior messages from the sender (**1803**) then routing information used in prior messages is retrieved (**1805**) in this example. Routing information may for example be extracted from the header in email. Routing information may for example be obtained during an instant message session by retrieving the IP address of the party being communicated with, for example as is done when an instant message session coordinates a peer-to-peer activity such as a direct file transfer with a remote party. In some embodiments, routing information may be stored implicitly, such as in headers of a previously received message, or explicitly, for example by extracting routing information and storing it in a database or file system. Routing information for the current message is compared to prior messages (**1806**), for example by checking that the IP address for the party being communicated with is in a similar subnet, or allocated within the same geographical region, for example the same continent, as a previous IP address for the same party. IP address to geographical location may be estimated by various software applications and services, such as services provided by MaxMind and described on http://www.maxmind.com. In some embodiments, routing information for an email message may be deduced by examining header information related to mail transfer agents. In some embodiments, similarity in IP addresses for mail transfer agents may be interpreted as implying similar routing. If routing is similar to previous messages from the same sender (**1806**), then the message is processed as legitimate (**1808**) in this example. If the routing information for a message purporting to be from a sender is not similar to previous messages from the same sender (**1806**) then the message is processed as suspicious (**1807**) in this example. In some embodiments, processing as suspicious may include contributing to a scoring that labels the message as an undesirable message, such as spam, viral, or phishing. In some embodiments, higher probability suspicious messages may be scrutinized further, or may be restricted to some extent in their impact on the computer. An example of restricting impact of a message may include discarding the message. Another example of restricting impact or further scrutiny of a message may include placing a message in a spam inbox. Another example of a restricting impact of a message may include preventing execution of content contained in the message, and/or requiring additional user interactions to bypass the restriction, for example by having a user agree to ignore warnings which describe the suspicious circumstances. In various embodiments, the process of FIG. **18** is implemented on a mail, messaging, or other server associated with the sender and/or the sender's ISP; a mail, messaging, or other server associated with the recipient and/or the recipient's ISP; and/or one or more intermediate nodes.

FIG. **19**A is a flow diagram of a method for notifying possible sources of infection according to some embodiments. In this example, an infected message is received (**1901**). An example of an infected message is a message classified as such by anti-contagion software. Routing information is extracted from the message (**1902**). An example of extracting routing information in an email message is to extract header lines inserted by mail transfer agents that propagated the message. Another example of extracting routing information is to acquire an IP address of a party making contact via instant messaging, for example as is provided by AOL's instant message infrastructure for a peer-to-peer file transfer. Routing information from previously received messages is retrieved (**1903**). An example of retrieving email routing information from prior messages is scanning all received messages, such as messages in an email inbox, and extracting routing information. Another example of retrieving routing information is to query a database or file system where routing information from previous messages has been gathered. Routing information for the infected message is compared with the retrieved routing information (**1904**). If no similarity with previous messages is found (**1904**) then in this example a third party notification is made (**1906**). An example of a third party notification is sending notification to the ISP

that provides network access to the message source. Another example of notifying a third party is notifying a viral clearinghouse of the apparent source of a viral message. Another example of a notifying a third party is notifying a service that determines the credibility of a report, such as a collaborative filter service, that accumulates accusations of viral disseminations. In some embodiments, similarity of routing of an infected message with a previous message may include having a previous message from the IP address of the infected message source. If a similarity is found with previous messages (**1904**) then in this example a notification is sent (**1905**). In some embodiments, notification may be sent (**1905**) to one or more sources of the prior message(s) that best match the infected message, and/or to third parties. As an example, notification may be sent to an email source that had the same originating IP address as the infected message. In some embodiments, a user may be asked to approve notification(s), for example via a pop-up dialog. In some embodiments, notification may be automatically sent without user intervention.

FIG. **19**B is a block diagram illustrating a network environment in which suspicious messages are identified and/or one or more notifications are sent when a malicious message is detected. A message source **1920** sends a message via the Internet **1922** to a message processing system **1926** for delivery via the Internet to one or more message destinations **1924**. In various embodiments, the message processing system **1926** comprises a mail or messaging server associated with the message source **1920**, a mail or messaging server associated with the destination(s) **1924**, and/or an intermediate processing node. In some embodiments, the message source **1920** is connected directly and/or via a local network, and not via the Internet **1922** as shown in FIG. **19**B, to message processing system **1926**. In some embodiments, message processing system **1926** is configured to identify suspicious messages using current and historic routing information, e.g., as described above in connection with FIG. **18**. In some embodiments, the process of FIG. **19**A is implemented on message processing system **1926**. For example, in some embodiments if message processing system **1926** determines that a message sent by message source **1920** is infected, message processing system **1926** compares routing information associated with the malicious message with corresponding information for one or more messages received previously from the message source **1920**. If the routing information is similar, in some embodiments the message processing system **1926** notifies the message source **1920**. If the routing information is not similar, in various embodiments the message processing system **1926** notifies the message source **1920**, the sender's ISP **1928**, and/or a clearinghouse or similar applicable service **1930**, as described above.

Although the foregoing embodiments have been described in some detail for purposes of clarity of understanding, the invention is not limited to the details provided. There are many alternative ways of implementing the invention. The disclosed embodiments are illustrative and not restrictive.

What is claimed is:

1. A method for protecting a network, comprising:

detecting an insecure condition on a first host that has connected or is attempting to connect to a protected network, wherein detecting the insecure condition includes contacting a trusted computing base associated with a trusted platform module within the first host, receiving a response, and determining whether the response includes a valid digitally signed attestation of cleanliness, wherein the valid digitally signed attestation of cleanliness includes at least one of an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation that the trusted computing base has ascertained the presence of a patch or a patch level associated with a software component on the first host;

when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantining the first host, including by preventing the first host from sending data to one or more other hosts associated with the protected network, wherein preventing the first host from sending data to one or more other hosts associated with the protected network includes receiving a service request sent by the first host, serving a quarantine notification page to the first host when the service request comprises a web server request, and in the event the service request comprises a DNS query, providing in response an IP address of a quarantine server configured to serve the quarantine notification page if a host name that is the subject of the DNS query is not associated with a remediation host configured to provide data usable to remedy the insecure condition; and

permitting the first host to communicate with the remediation host.

2. A method as recited in claim **1**, wherein detecting an insecure condition further includes at least one of the following: scanning for a vulnerability, scanning for malicious data, determining whether a security software is installed, and detecting anomalous network traffic.

3. A method as recited in claim **1**, wherein detecting an insecure condition includes determining that the first host should be quarantined until an update to an operating system has been installed.

4. A method as recited in claim **1**, wherein detecting an insecure condition includes configuring an operating system to quarantine the first host upon initial startup after installation of the operating system.

5. A method as recited in claim **1**, wherein permitting the first host to communicate with the remediation host includes:

detecting an outbound communication from the first host; and

forwarding the outbound communication if it is addressed to the remediation host.

6. A method as recited in claim **1**, wherein preventing the first host from sending data to the one or more other hosts includes:

detecting an outbound communication from the first host; and

redirecting the outbound communication to a quarantine server if it comprises a request for an approved service and is not addressed to a remediation host.

7. A method as recited in claim **1**, wherein quarantining the first host further includes preventing the first host from receiving via the protected network data not related to remediation of the insecure condition.

8. A method as recited in claim **1**, performed at an Internet service provider.

9. A method as recited in claim **1**, wherein the software component on the first host is an operating system.

10. A method as recited in claim **1**, wherein determining that the response does not include a valid digitally signed attestation of cleanliness includes determining that the software component on the first host is not sufficiently updated.

11. A method as recited in claim **1**, wherein determining that the software component on the first host is not sufficiently updated includes determining that a patch level associated with the software component on the first host is not sufficiently recent.

12. A system for protecting a network, comprising:

a processor configured to:

detect an insecure condition on a first host that has connected or is attempting to connect to a protected network, wherein detecting the insecure condition includes contacting a trusted computing base associated with a trusted platform module within the first host, receiving a response, and determining whether the response includes a valid digitally signed attestation of cleanliness, wherein the valid digitally signed attestation of cleanliness includes at least one of an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation that the trusted computing base has ascertained the presence of a patch or a patch level associated with a software component on the first host;

when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantine the first host, including by preventing the first host from sending data to one or more other hosts associated with the protected network, wherein preventing the first host from sending data to one or more other hosts associated with the protected network includes receiving a service request sent by the first host, serving a quarantine notification page to the first host when the service request comprises a web server request, and in the event the service request comprises a DNS query, providing in response an IP address of a quarantine server configured to serve the quarantine notification page if a host name that is the subject of the DNS query is not associated with a remediation host configured to provide data usable to remedy the insecure condition; and

permit the first host to communicate with the remediation host; and

a memory coupled to the processor and configured to provide instructions to the processor.

13. A system as recited in claim 12, wherein the processor is configured to detect an insecure condition at least in part by performing one or more of the following: scanning for a vulnerability, scanning for malicious data, determining whether a security software is installed, and detecting anomalous network traffic.

14. A system as recited in claim 12, wherein the processor is configured to detect an insecure condition at least in part by determining that an initial startup after installation of an operating system is being performed.

15. A system as recited in claim 12, wherein the processor is configured to quarantine the first host at least in part by preventing the first host from receiving via the protected network data not related to remediation of the insecure condition.

16. A system as recited in claim 12, wherein the software component on the first host is an operating system.

17. A system as recited in claim 12, wherein determining that the response does not include a valid digitally signed attestation of cleanliness includes determining that the software component on the first host is not sufficiently updated.

18. A system as recited in claim 17, wherein determining that the software component on the first host is not sufficiently updated includes determining that a patch level associated with the software component on the first host is not sufficiently recent.

19. A computer program product for protecting a network, the computer program product being embodied in a non-transitory computer readable medium and comprising computer instructions for:

detecting an insecure condition on a first host that has connected or is attempting to connect to a protected network, wherein detecting the insecure condition includes contacting a trusted computing base associated with a trusted platform module within the first host, receiving a response, and determining whether the response includes a valid digitally signed attestation of cleanliness, wherein the valid digitally signed attestation of cleanliness includes at least one of an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation that the trusted computing base has ascertained the presence of a patch or a patch level associated with a software component on the first host;

when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantining the first host, including by preventing the first host from sending data to one or more other hosts associated with the protected network, wherein preventing the first host from sending data to one or more other hosts associated with the protected network includes receiving a service request sent by the first host, serving a quarantine notification page to the first host when the service request comprises a web server request, and in the event the service request comprises a DNS query, providing in response an IP address of a quarantine server configured to serve the quarantine notification page if a host name that is the subject of the DNS query is not associated with a remediation host configured to provide data usable to remedy the insecure condition; and

permitting the first host to communicate with the remediation host.

* * * * *

EXHIBIT 5

SHERIDAN ROSS pc attorneys at innovation

January 30, 2025

Mr. Tony Gomes
Cloud Software Group, Inc.
Chief Legal & Administrative Officer
Citrix Business Unit
851 W. Cypress Creek Road
Fort Lauderdale, FL 33309

Robert R. Brunelli
Shareholder
Direct: 303.863.2980
rbrunelli@sheridanross.com

Via FedEx

Re: K.Mizra LLC

Dear Mr. Gomes:

As you may be aware, K.Mizra LLC ("K.Mizra") is a patent licensing company run by experienced management. The Company focuses on high value, high quality patents with a global reach. It owns patent portfolios originating with a wide array of inventors, including portfolios developed by well-known multinationals such as IBM, Intel, Rambus and others, as well as from research institutes such as Nederlandse Organisatie voor Toegespast Natuurwetenschappelijk Onderzoek. By focusing on high quality patents, K.Mizra provides a secondary market for inventors to recoup their research and development investments and to continue their innovations. K.Mizra offers licenses to its patents on reasonable terms and in this way plays an important part in the development of the technologies that improve all our lives.

K.Mizra is the owner by assignment of United States Patent Nos. 8,234,705 ("the '705 Patent") (Ex. 1) and 9,516,048 ("the '048 Patent") (Ex. 2). The '705 Patent, titled "Contagion Isolation and Inoculation," was duly and legally issued by the United States Patent and Trademark Office ("USPTO") and expires on November 25, 2029. The '048 Patent, titled "Contagion Isolation and Inoculation Via Quarantine," was issued by the USPTO December 6, 2016 and expires on September 27, 2025.

The technology described and claimed in these Patents was invented by Dr. James Roskind and Mr. Aaron Emigh, two prolific inventors and colleagues living in the same area who had similar interests in innovating computer-related technologies. Dr. Roskind has bachelors, masters, and doctorate degrees from MIT in both electrical engineering and computer science and is the named inventor of over 300 U.S. patents. He has worked for Netscape as the Chief Architect and as the Netcenter Security Architect, and was a co-founder for Infoseek, a company that was eventually acquired by Disney for $770 million. He was also a key developer of Google's "transport protocol" that provides the tech giant billions of dollars in value every year. Mr. Emigh graduated from the University of California, Santa Cruz with degrees in linguistics and computer and information sciences and is the named inventor of over 140 patents. Prior to working with Dr. Roskind, Mr. Emigh worked in various positions developing software, including working as a software manager, architect, and engineer for Unicom and working as a manager for the software development and technical marketing groups for Philips TriMedia. He has founded or co-founded many companies, including CommerceFlow, Inc., which was acquired by eBay for its technology that Mr. Emigh helped to develop.

In 2003, the inventors decided to create a business together, Radix Labs, LLC, which focused on developing intellectual property related to various computer technologies, including network security technologies. The inventors focused on conceiving inventions they knew were needed then or soon would be needed in the computer networking industry and then on drafting patent applications to capture and protect their innovations. In September of 2004, the inventors filed the Provisional Application to which both Patents claim priority. The Provisional Application described technology that focused on securing a network against the threats to which it was exposed by the connections of additional end points to that network. The Provisional Application, and by natural extension the Patents, also focus on remedying identified threats and quarantining those threats to mitigate any damage to the secured network.

More specifically, '705 Patent explains that "[l]aptop and wireless computers and other mobile systems pose a threat to elements comprising and/or connected to a network service provider, enterprise, or other protected network to which they reconnect after a period of connection to one or more networks and/or systems that are not part of the service provider, enterprise, or other protected network. By roaming to unknown domains, such as the Internet, and/or connecting to such domains through public, wireless, and/or otherwise less secure access nodes, such mobile systems may become infected by computer viruses, worms, backdoors, and/or countless other threats and/or exploits and/or have unauthorized software installed; have software installed on the mobile system by an operator of the protected network for the protection of the mobile system and/or the protected network removed or altered without authorization and/or have configurations, settings, security data, and/or other data added, removed, and/or changed in authorized ways and/or by unauthorized person[s]." *See, e.g.*, Ex. 1 at 1:14–31. The solution provided by and embodied in the Asserted Patents to these problems was a complete break from the conventional network access control solutions then in use, required unique hardware to be properly implemented and was then, as today, patent-subject eligible, highly valuable, novel and non-obvious technology.

After the Patents issued, Dr. Roskind and Mr. Emigh recouped their research and development investment by selling their rights thereto, continuing in their individual technology development pursuits. K.Mizra ultimately acquired these Patents and licensed them to many of the who's-who of the tech world. Some of the accused infringers chose to test the validity of the Patents prior to settling with K.Mizra. For instance, a few accused infringers previously sought *inter partes* review ("IPR") by the Patent Trial and Appeal Board ("PTAB") of the Patents. A Final Written Decision ("Decision") in the '705 Patent IPR found that the petitioners had not shown, by a preponderance of the evidence, that the asserted claims were unpatentable. Based on the '705 Patent IPR Decision, the similar '048 Patent IPR was not even instituted. The '705 Patent IPR Decision was appealed to the Court of Appeals for the Federal Circuit ("CAFC"), resulting in a procedurally focused remand back to the PTAB. That appeal is now concluded and the parties to the IPR have recently filed with the Board a joint motion to dismiss.

Cloud Software Group Inc. ("CSG") is in the business of providing VPN-focused security products and services, among others, to its customers in the U.S. and elsewhere. Included within these offerings is Citrix Zero Trust Secure Access ("Involved Product"). As can be seen through review of the attached Notice of Technology Use chart (Ex. 3), the Involved Product meets all the limitations of several of the Patents' claims. CSG's activities now need to be regularized, with

K.Mizra remaining ready, willing and able to provide CSG a commercially reasonable license to its various patented technologies, including the Patents. If you are interested in discussing purchasing such a license, please contact me directly.

Thank you for your attention to this matter.

Very truly yours,

SHERIDAN ROSS P.C.

Robert R. Brunelli
RRB/lrb
Enclosures
C: K.Mizra LLC

EXHIBIT 6

December 5, 2024

Contributed by:

IN THIS ARTICLE

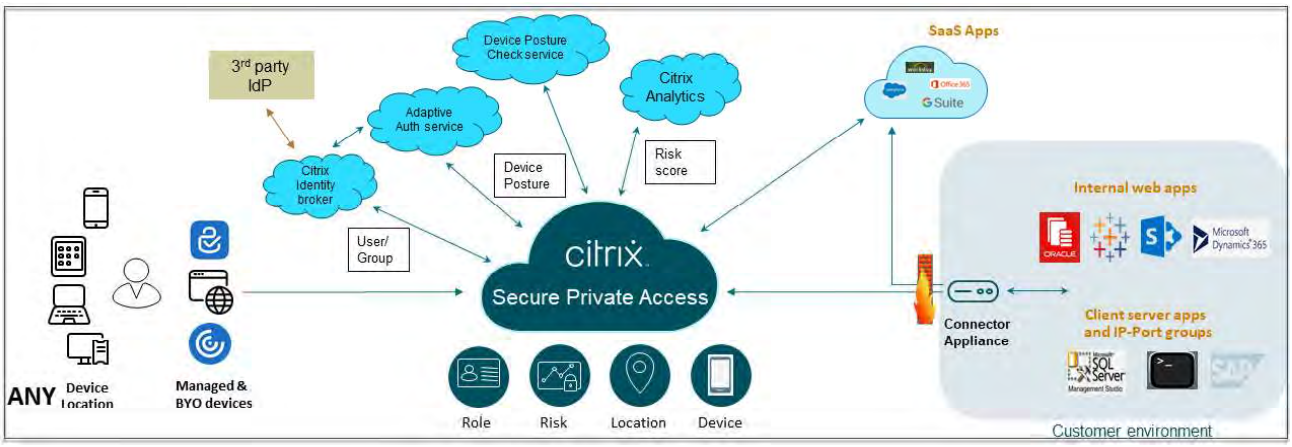
Solution overview

- Adaptive security
- Application access
- Single sign-on
- Browser security
- Device posture

Solution overview

Traditional VPN solutions require end-user devices to be managed, provide access at the network level, and enforce static access control policies. Citrix Secure Private Access gives IT a set of security controls to protect against threats from BYO devices, giving users the choice to access their IT-sanctioned applications from any device, whether it's managed or BYO.

Citrix Secure Private Access offers Adaptive Authentication, single sign-on support, enhanced security controls for the applications. Secure Private Access also provides the capabilities to scan the end user device before establishing a session by using the Device Posture service. Based on the Adaptive Authentication or Device Posture results, admins can define the authentication methods for the apps.



Adaptive security

Adaptive Authentication determines the right authentication flow for the current request. Adaptive Authentication can identify the device posture, geographical location, network segment, user organization/department membership. Based on the information obtained, an admin can define how they want to authenticate users to their IT sanctioned apps. This allows organizations to implement the same authentication policy framework across every resource including public SaaS apps, private web apps, private client-server apps, and Desktops as a Service (DaaS). For details, see Adaptive Security.

K.Mizra LLC, Ex. 2001, p. 88
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

Application access

Secure Private Access can create a connection to the on-premises web apps without relying on a VPN. This VPN-less connection uses an on-premises deployed Connector Appliance. The Connector Appliance creates an outbound control channel to the organization's Citrix Cloud subscription. From there, Secure Private Access can tunnel connections to the internal web apps without the need for a VPN. For details, see **Application Access**.

Single sign-on

With Adaptive Authentication, organizations can provide strong authentication policies to help reduce the risk of compromised user accounts. The single sign-on capabilities of Secure Private Access use the same Adaptive Authentication policies for all SaaS, private web, and client-server apps. For details, see **Single Sign-On**.

Browser security

Secure Private Access enables end users to safely browse the internet with a centrally managed and secured enterprise browser. When an end user launches a SaaS or private web app, several decisions are dynamically made to decide how best to serve this application. For details, see **Browser Security**.

Device posture

Device posture service allows an admin to define policies to check the posture of endpoint devices trying to access corporate resources remotely. Based on the compliance status of an endpoint, the device posture service can deny access or provide restricted/full access to corporate applications and desktops.

When an end user initiates a connection with Citrix Workspace, the Device Posture client collects information about the endpoint parameters and shares this information with the Device Posture service to determine if the posture of the endpoint meets policy requirements.

The integration of the Device Posture service with Citrix Secure Private Access enables secure access to SaaS, Web, TCP and UDP apps from anywhere, delivered with the resiliency and scalability of Citrix Cloud. For details, see **Device Posture**.

Support for TCP and UDP applications

Sometimes remote users need access to private client-server apps that have their front-end on the endpoint and their back-end in a data center. Organizations can rightfully enforce strict security policies around these internal and private apps, making it difficult for remote users to access these applications without compromising security protocols.

Secure Private Access service addresses the TCP and UDP security vulnerabilities by enabling ZTNA to deliver secure access to these apps. Users can now access all private apps including TCP, UDP, and HTTPS apps either using a native browser or a native client application via the Citrix Secure Access client running on their machines.

Users must install the Citrix Secure Access client on their client devices.

- For Windows, the client version (22.3.1.5 and later) can be downloaded from **https://www.citrix.com/downloads/citrix-gateway/plug-ins/citrix-secure-access-client-for-windows.html**.
- For macOS, the client version (22.02.3 and later) can be downloaded from the App Store.

For details, see **Support for client-server apps**.

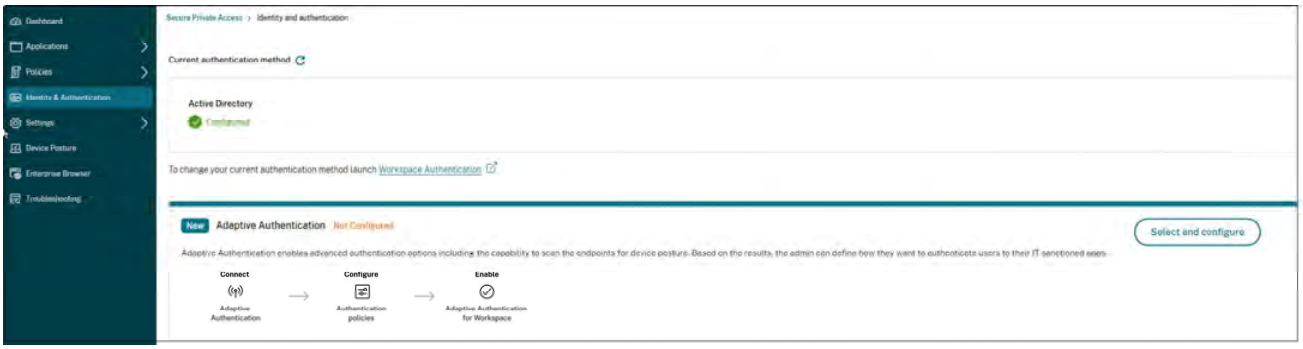
Set up Citrix Secure Private Access

Enable zero trust network access to SaaS apps, internal web apps, TCP, and UDP apps using the Secure Private Access admin console. This console includes configuration of Adaptive Authentication, applications including user subscription and adaptive access policies.

K.Mizra LLC, Ex. 2001, p. 89
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

Set up identity and authentication

Select the authentication method for the subscribers to log in to Citrix Workspace. Adaptive Authentication is a Citrix Cloud service that enables advanced authentication for customers and users logging in to Citrix Workspace.



For details, see Set up identity and authentication.

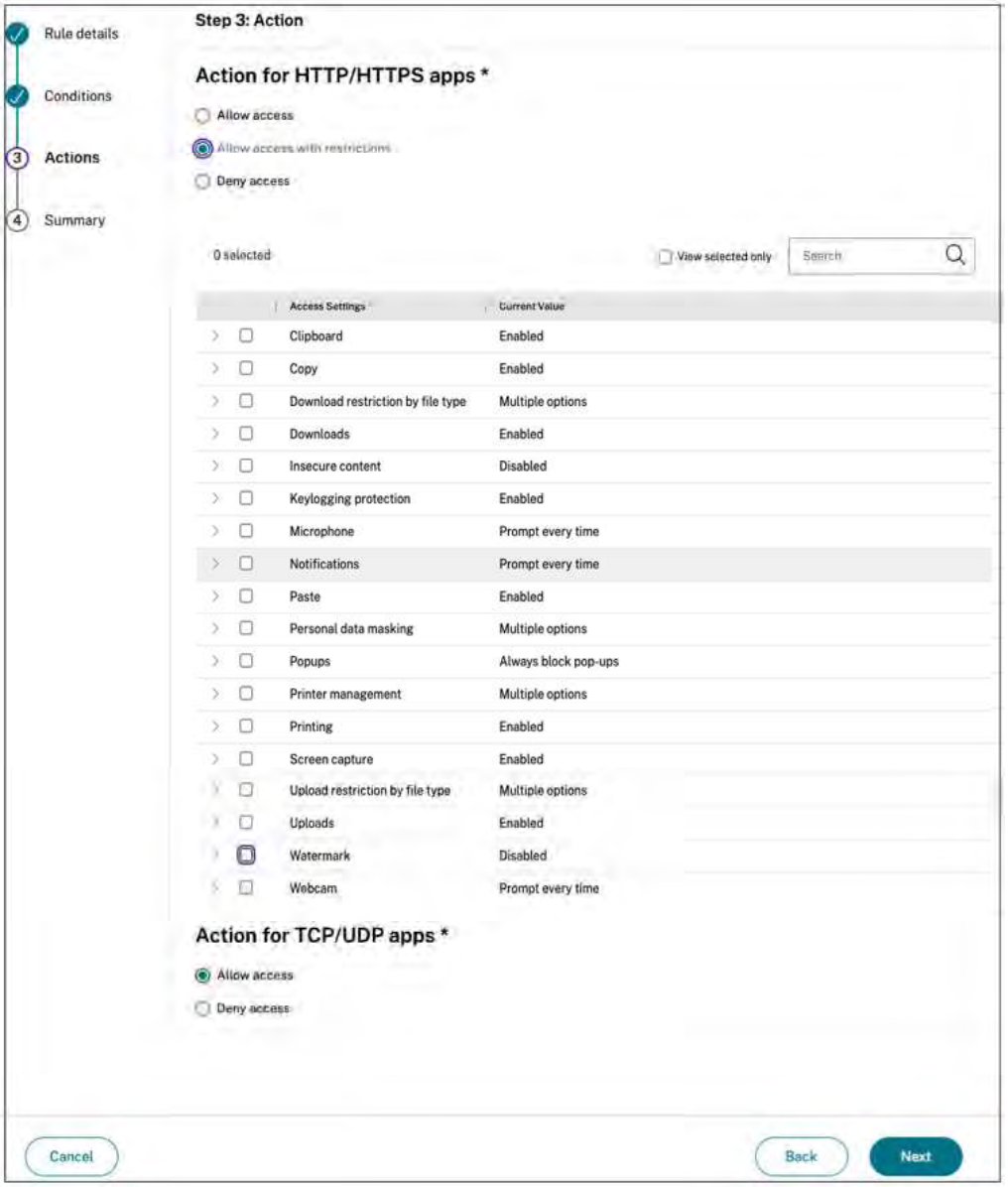
Enumerate and publish apps

After you have selected the authentication method, configure the Web, SaaS, or the TCP and UDP apps using the admin console. For details, see Add and manage apps.

Enable enhanced security controls

To protect content, organizations incorporate enhanced security policies within the SaaS applications. Each policy enforces a restriction on the Citrix Enterprise Browser when using Workspace app for desktop or on Secure Browser when using Workspace app web or mobile.

- **Restrict clipboard access**: Disables cut/copy/paste operations between the app and the system clipboard.
- **Restrict printing**: Disables the ability to print from within the Citrix Enterprise Browser.
- **Restrict downloads**: Disables the user's ability to download from within the app.
- **Restrict uploads**: Disables the user's ability to upload within the app.
- **Display watermark**: Displays a watermark on the user's screen displaying the user name and IP address of the user's machine.
- **Restrict key logging**: Protects against key loggers. When a user tries to log on to the app using the user name and password, all the keys are encrypted on the key loggers. Also, all activities that the user performs on the app are protected against key logging. For example, if app protection policies are enabled for Office 365 and the user edit an Office 365 word document, all key strokes are encrypted on key loggers.
- **Restrict screen capture**: Disables the ability to capture the screens using any of the screen capture programs or apps. If a user tries to capture the screen, a blank screen is captured.



For details, see Configure an access policy.

Enable Citrix Enterprise Browser for application launches

Secure Private Access enables end users to launch their apps using the Citrix Enterprise Browser (CEB). CEB is a chromium-based browser integrated with the Citrix Workspace app that enables a seamless and secure access experience to access web and SaaS apps within Citrix Enterprise Browser.

CEB can be configured as preferred browser or as your work browser for all the internally hosted web apps or SaaS apps with security policies. CEB allows users to open all configured SaaS/web app domains inside a secure and controlled environment.

Enable Citrix Enterprise Browser

Administrators can use Global App Configuration service (GACS) to configure Citrix Enterprise Browser as the default browser to launch web and SaaS apps from the Citrix Workspace app.

Configuration through API:

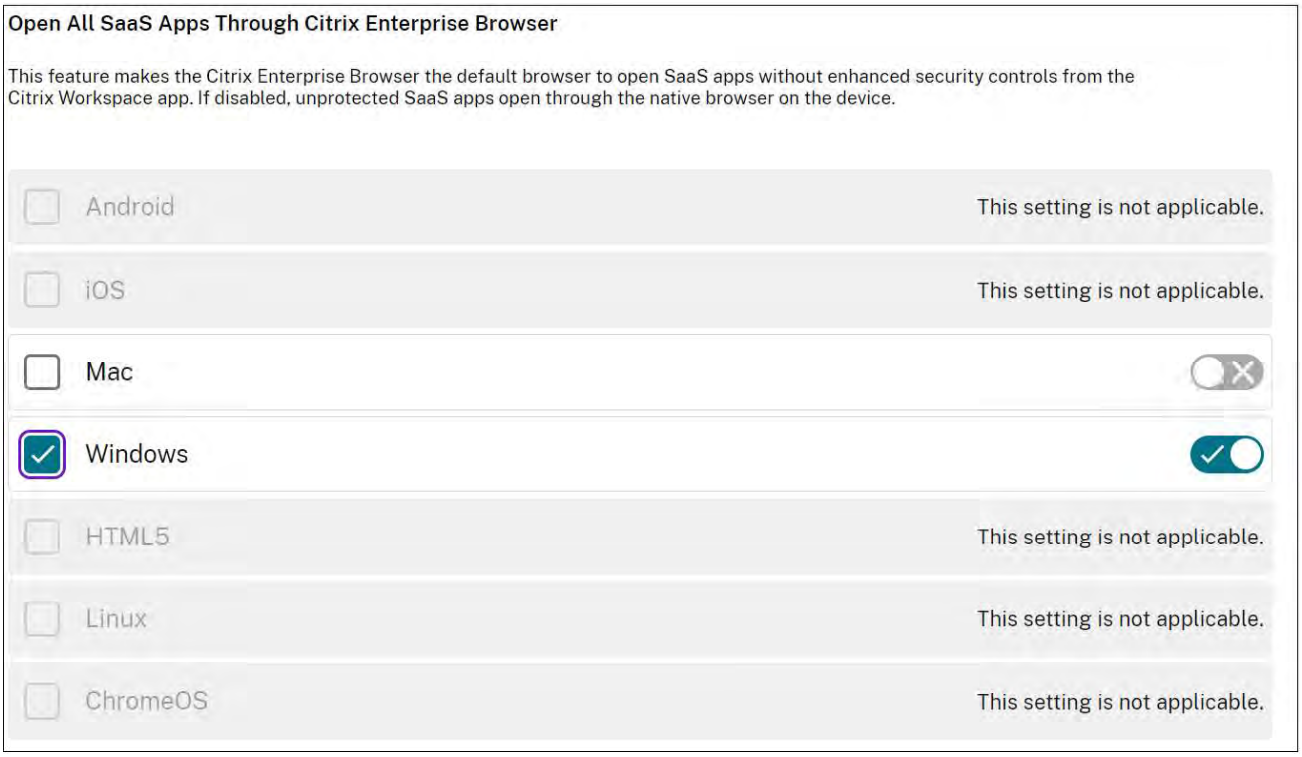
To configure, here is an example JSON file to enable Citrix Enterprise Browser for all apps, by default:

```
"settings": [
                {
                   "name": "open all apps in ceb",
                   "value": "true"
                }
                ]
```

The default value is true.

Configuration through GUI:

Select the devices for which CEB must be made the default browser for the app launches.



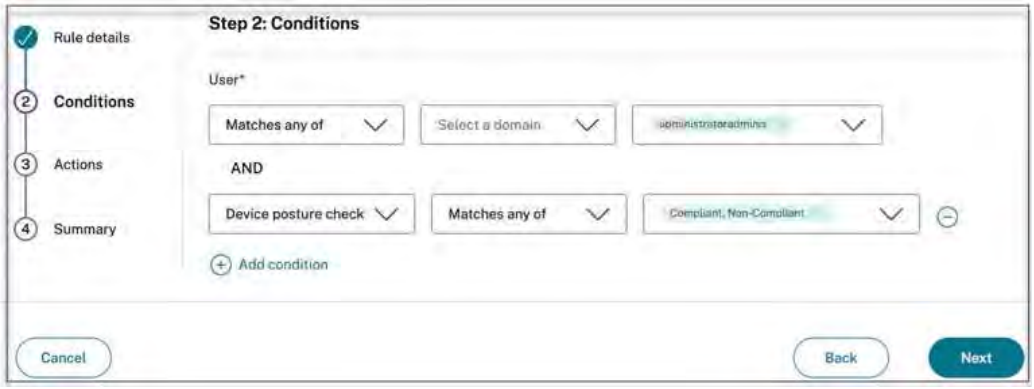
For details, see Manage Citrix Enterprise Browser through GACS.

Configure tags for contextual access using Device Posture

After the device posture verification, the device is allowed to log in and the device is classified as compliant or non-compliant. This classification is made available as tags to the Secure Private Access service and are used to provide contextual access based on device posture.

1. Sign into Citrix Cloud.
2. On the Secure Private Access tile, click **Manage**.
3. Click **Access Policies** on the left navigation and then click **Create policy**.

4. Enter the policy name and description of the policy.
5. In **Applications**, select the app or set of apps on which this policy must be enforced.
6. Click **Create Rule** to create rules for the policy.
7. Enter the rule name and a brief description of the rule, and then click **Next**.
8. Select the users' conditions. The Users condition is a mandatory condition to be met to grant access to the applications for the users.
9. Click **+** to add device posture condition.
10. Select **Device posture check** and the logical expression from the drop-down menu.
11. Enter one of the following values in custom tags:



- **Compliant** - For compliant devices
- **Non-Compliant** - For non-compliant devices

12. Click **Next**.
13. Select the actions that must be applied based on the condition evaluation, and then click **Next**.

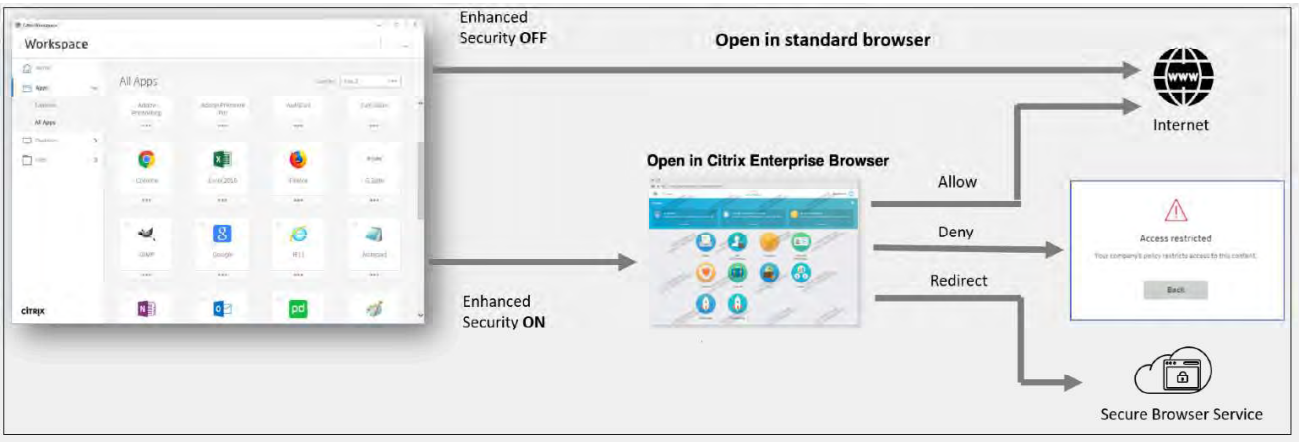
    The Summary page displays the policy details.
14. Verify the details and click **Finish**.

NOTE:

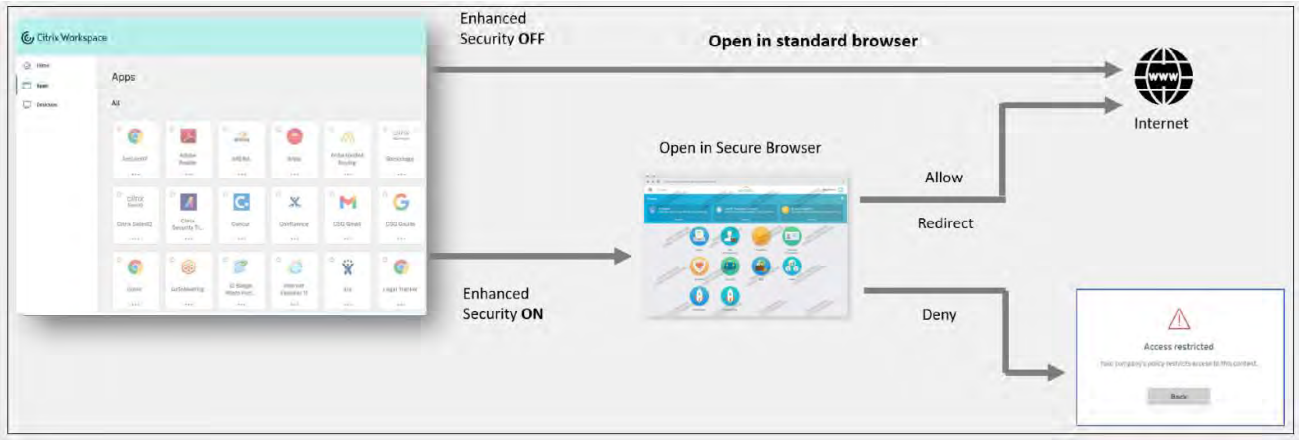
Any Secure Private Access application which is not tagged as compliant or non-compliant in the access policy is treated as the default application and is accessible on all the endpoints regardless of device posture.

End-user experience

The Citrix administrator has the power to extend security control with the help of Citrix Secure Private Access. Citrix Workspace app is an entry point to access all resources securely. End users can access virtual apps, desktops, SaaS apps, and files through Citrix Workspace app. With Citrix Secure Private Access, administrators can control how a SaaS Application is accessed by the end user via Citrix Workspace Experience web UI or native Citrix Workspace app client.



When the user launches the Workspace app on the endpoint, they see their applications, desktops, files, and SaaS apps. If a user clicks the SaaS application when enhanced security is disabled, the application opens in a standard browser which is locally installed. If the administrator has enabled enhanced security, then the SaaS apps open on the CEB within the Workspace app. Accessibility to hyperlinks within SaaS apps and web apps is controlled based on the unsanctioned websites policies. For details on Unsanctioned websites, see Unsanctioned websites.



Similarly, with the Workspace Web portal, when enhanced security is disabled, SaaS applications are opened in a standard browser which is natively installed. When enhanced security is enabled, SaaS apps are opened in the secure Remote Browser. Users can access the websites within SaaS apps based on the unsanctioned websites policies. For details on Unsanctioned websites, see Unsanctioned websites.

Analytics dashboard

The Secure Private Access service dashboard displays the diagnostics and usage data of the SaaS, Web, TCP, and UDP apps. The dashboard provides admins full visibility into their apps, users, connectors health status, and bandwidth usage in a single place for consumption. This data is fetched from Citrix Analytics. The metrics are broadly classified into the following categories.

- Logging and troubleshooting
- Users
- Applications
- Access policies

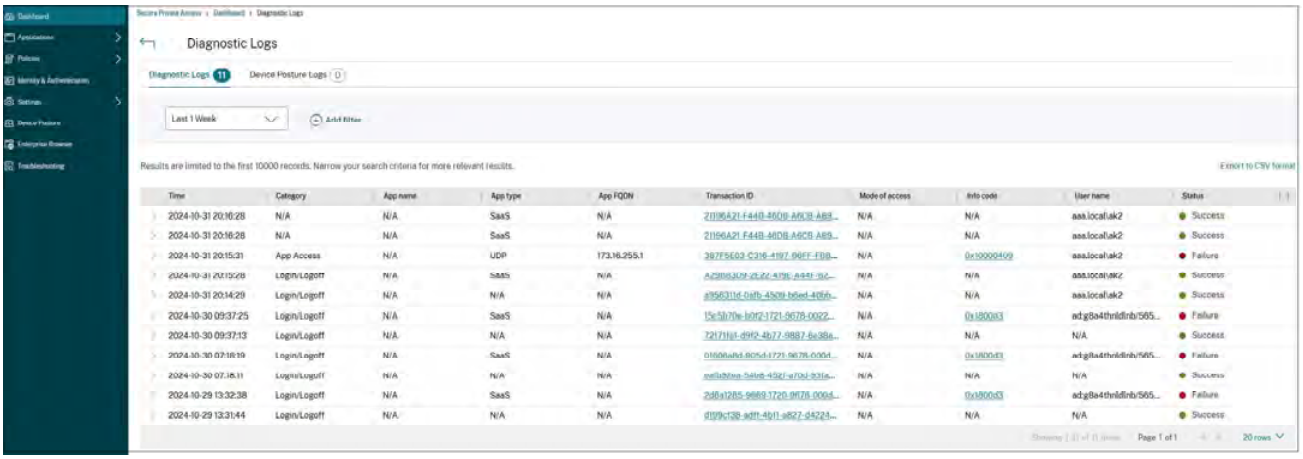
For details, see Dashboard.



Troubleshoot app issues

The Diagnostics Logs chart in the Secure Private Access dashboard provides visibility into the logs related to authentication, application launch, app enumeration, and device posture logs.

- **Info code**: Some log events such as failures have an associated info code. Clicking the info code redirects the users to the resolution steps or more information about that event.
- **Transaction ID**: The diagnostic logs also display a transaction ID that correlates all Secure Private Access logs for an access request. One app access request can have multiple logs generated, starting from authentication, then app enumeration within the workspace app, and then app access itself. All these events generate their own logs. Transaction ID is used to correlate all of these logs. You can filter the diagnostic logs using the transaction ID to find all logs related to a particular app access request. For details, see Troubleshoot Secure Private Access issues.



Sample use cases

- Access internal applications (Web/TCP/UDP) using a Zero-Trust approach without opening incoming traffic on the firewall

K.Mizra LLC, Ex. 2001, p. 95
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

- Move to a Zero-Trust approach by discovering applications accessed by users
- Restrict access to SaaS applications to Citrix Enterprise Browser
- Restrict access to SaaS applications to company-owned public IP addresses
- Enhanced Security to Azure-managed SaaS Apps
- Enhanced Security to Office 365
- Enhanced Security to Okta Apps

Reference articles

- Introduction to Secure Private Access
- Tech brief
- Reference Architecture
- Citrix Enterprise Browser
- Manage Citrix Enterprise Browser through GACS
- Admin-guided workflow for easy onboarding and set up

Reference videos

- Zero trust network access (ZTNA) to apps
- Private Web app access with Citrix Secure Private Access
- Public SaaS app access with Citrix Secure Private Access
- Private client-server app access with Citrix Secure Private Access
- Keylogger Protection with Citrix Secure Private Access
- Screen sharing protection with Citrix Secure Private Access
- End-user experience with Citrix Secure Private Access
- ZTNA versus VPN logon experience with Citrix Secure Private Access
- ZTNA versus VPN port scans with Citrix Secure Private Access

What's new in related products

- Citrix Enterprise Browser: About this release
- Citrix Workspace: What's new
- Citrix DaaS: What's new
- Citrix Secure Access client NetScaler Gateway Clients

Was this helpful

Send us your feedback

Instructions for Contributors

citrix

Site feedback | Your Privacy Choices | Privacy and legal terms | Cookie preferences | docs.cloud.com

© 1999-2025 Cloud Software Group, Inc. All rights reserved.

EXHIBIT 7

March 3, 2025

Contributed by:

IN THIS ARTICLE

Prerequisites

How it works

Scans supported by the Device Posture service

Third-party integration with the Device Posture service

Known limitations

Citrix Device Posture service is a cloud-based solution that helps admins to enforce certain requirements that the end devices must meet to gain access to Citrix DaaS (virtual apps and desktops) or Citrix Secure Private Access resources (SaaS, Web apps, TCP, and UDP apps). Establishing device trust by checking the device's posture is critical to implement zero-trust-based access. Device Posture service enforces zero trust principles in your network by checking the end devices for compliance (managed/BYOD and security posture) before allowing an end user to log in.

Prerequisites

- Licensing requirements: The Device Posture service is available as part of the Universal Hybrid Multi Cloud License (UHMC) and Platform License (CPL). For more information, see https://www.citrix.com/buy/licensing/product.html.
- Supported platforms:
  - Windows (10 and 11)
  - macOS 13 Ventura
  - macOS 12 Monterey
  - iOS
  - IGEL
  - Linux

K.Mizra LLC, Ex. 2001, p. 99
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

NOTE:

- A device running on a non-supported platform is marked as non-compliant by default. You can change the classification from **Non-compliant** to **Denied login** from the **Settings** tab on the Device Posture page. For the definitions of "compliant" and "non-compliant," see Definitions.
- A device that is running on a supported platform but does not match any pre-defined device posture policy is marked as non-compliant, by default. You can change the classification from **Non-compliant** to **Denied login** from the **Settings** tab on the Device Posture page.
- For iOS support on Device Posture service, the EPA client is built in as part of the Citrix Workspace app for iOS. For details on the versions, see Citrix Workspace app for iOS.
- For IGEL OS support on the Device Posture service, the EPA client is built in as part of the IGEL OS. Contact the IGEL support team for installing the EPA client on the IGEL devices.

- Citrix Device Posture client (EPA client): A lightweight application that must be installed on the endpoint device to run device posture scans. This application does not require local admin rights to download and install on an endpoint.

NOTE:

If you're using a device certificate check, then you must install the EPA client with administrative rights.

- Supported browsers: Chrome, Edge, and Firefox.
- Firewall configuration: To allow the Device Posture service to update the EPA clients on an end device, the firewall/proxy must be configured to allow the following domains:
  - `https://swa-ui-cdn-endpoint-prod.azureedge.net`
  - `https://productioniconstorage.blob.core.windows.net`
  - *.netscalergateway.net
  - *.nssvc.net
  - *.cloud.com
  - *.pendo.io
  - *.citrixworkspacesapi.net

How it works

The admins can create device posture policies to check the posture of endpoint devices and determine whether an endpoint device is allowed or denied login. The devices, which are allowed to log in are further classified as compliant or non-compliant. Users can log in from a browser or the Citrix Workspace app.

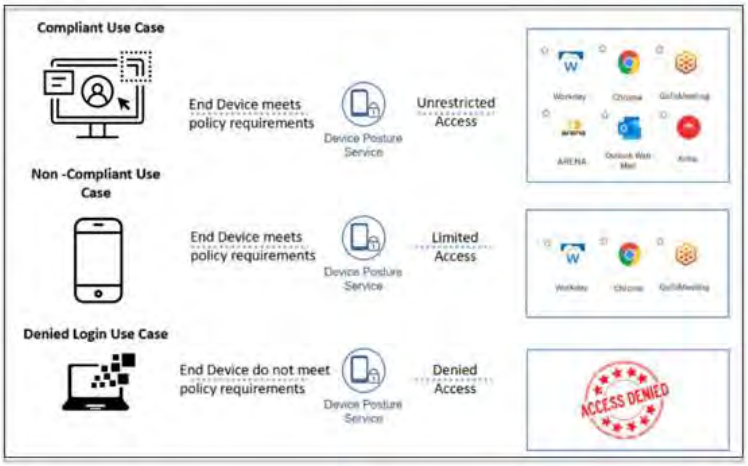
Following are the high-level conditions used to classify a device as compliant, non-compliant, and denied login.

- **Compliant devices** – A device that meets the pre-configured policy requirements and is allowed to log in into the company's network with full or unrestricted access to Citrix Secure Private Access resources or Citrix DaaS resources.
- **Non-Compliant devices** - A device that meets the pre-configured policy requirements and is allowed to log in into the company's network with partial or restricted access to Citrix Secure Private Access resources or Citrix DaaS resources based on your DaaS and Secure Private Access configuration for Device Posture.

- DaaS: Administrators can specify the delivery groups to restrict access for non-compliant devices. For details on how to configure restricted access to delivery groups, see Citrix DaaS configuration with Device Posture.
- Secure Private Access: Administrators can choose specific applications to restrict access for these devices. For details on how to restrict access to specific applications, see Citrix Secure Private Access configuration with Device Posture.

- **Denied login:** - A device that fails to meet the policy requirements is denied login.

The classification of devices as **compliant**, **non-compliant**, and **denied login** is passed onto the Citrix DaaS and Citrix Secure Private Access service that in turn uses the device classification to provide smart access capabilities.



NOTE:

- The device posture policies must be configured specifically for each platform. For example, for macOS, an admin can allow access for the devices that have a specific OS version. Similarly, for Windows, the admin can configure policies to include a specific authorization file, registry settings, and so on.
- Device posture scans are done only during pre-authentication/before logging in.
- For definitions of “compliant” and “non-compliant,” see Definitions.

Scans supported by the Device Posture service

The following scans are supported by the Citrix Device Posture service:

Windows	macOS	iOS	IGEL	Linux
Citrix Workspace App Version	Citrix Workspace App Version	Citrix Workspace App Version	-	-
OS Version	OS Version	OS Version	-	OS Version
File (Exists, Last Modified Time, MD5)	File (Exists, Last Modified Time, MD5)	-	File (Exists, Last Modified Time, MD5)	File (Exists, Last Modified Time, MD5)
Geo Location	Geo Location	-	-	-
Network Location	Network Location	-	-	-
MAC Address	MAC Address	-	-	-

Windows	macOS	iOS	IGEL	Linux
Process (Exists, MD5, SHA1)	Process (Exists, MD5, SHA1)	-	-	Process (Exists, MD5, SHA1)
Microsoft Endpoint Manager	Microsoft Endpoint Manager	-	-	-
CrowdStrike	CrowdStrike	-	-	-
Device Certificate	Device Certificate	-	-	-
Web Browser	Web Browser	-	-	-
Antivirus	Antivirus	-	-	-
Non-Numeric Registry (32 Bit)	-	-	-	-
Non-Numeric Registry (64 Bit)	-	-	-	-
Numeric Registry (32 Bit)	-	-	-	-
Numeric Registry (64 Bit)	-	-	-	-
Windows Update Installation Type	-	-	-	-
Windows Update Last Update Check	-	-	-	-
External Device Connected	-	-	-	-
Network Settings	-	-	-	-
-	-	-	-	Mount Point

NOTE:

- Starting from EPA library version 24.9.1.1, the **Windows Update Last Update** scan also includes updates installed through BigFix, Microsoft Intune, and other third-party tools, in addition to those installed through the Windows Auto Upgrade service.
- The **Windows Update Last Update** scan is not applicable for manually installed updates where the update patch is downloaded and installed by the user.
- For iOS support on Device Posture service, the EPA client is built in as part of the Citrix Workspace app for iOS. For details on the versions, see Citrix Workspace app for iOS.

Third-party integration with the Device Posture service

In addition to the native scans offered by the Device Posture service, the service can also be integrated with the following third-party solutions on Windows and macOS.

- Microsoft Intune. For details, see Microsoft Intune integration with Device Posture.
- CrowdStrike. For details, see CrowdStrike integration with Device Posture.

Known limitations

- The time taken for the device posture functionality to be enabled or disabled after the device posture toggle button is turned On or Off can take a few minutes to an hour.
- Any changes in the device posture configuration do not take effect immediately. It might take around 10 minutes for the changes to take effect.
- If you have enabled the Service Continuity option in Citrix Workspace and if the Device Posture service is down, users might be unable to sign in to Workspace. This is because Citrix Workspace enumerates apps and desktops based on local cache on the user device.
- If you have configured a long-lived token and password on Citrix Workspace, the device posture scan does not work for this configuration. The devices are scanned only when the users log in to Citrix Workspace.
- Each platform can have a maximum of 10 policies and each policy can have a maximum of 10 rules.
- Role-based access is not supported with the Device Posture service.
- The Device Posture service is not supported in Citrix Cloud Japan.

Quality of service

- Performance: Under ideal conditions, the Device Posture service adds an additional 2 seconds of delay during login. This delay might increase depending on additional configurations such as third-party integrations like Microsoft Intune.
- Resiliency: Device Posture service is highly resilient with multiple POPs to ensure that there's no downtime.

Definitions

The terms "compliant" and "non-compliant" in reference to the Device Posture service are defined as follows.

- **Compliant devices** – A device that meets the pre-configured policy requirements and is allowed to log into the company's network with full or unrestricted access to Citrix Secure Private Access resources or Citrix DaaS resources.
- **Non-Compliant devices** - A device that meets the pre-configured policy requirements and is allowed to log in into the company's network with partial or restricted access to Citrix Secure Private Access resources or Citrix DaaS resources.

Was this helpful

Send us your feedback

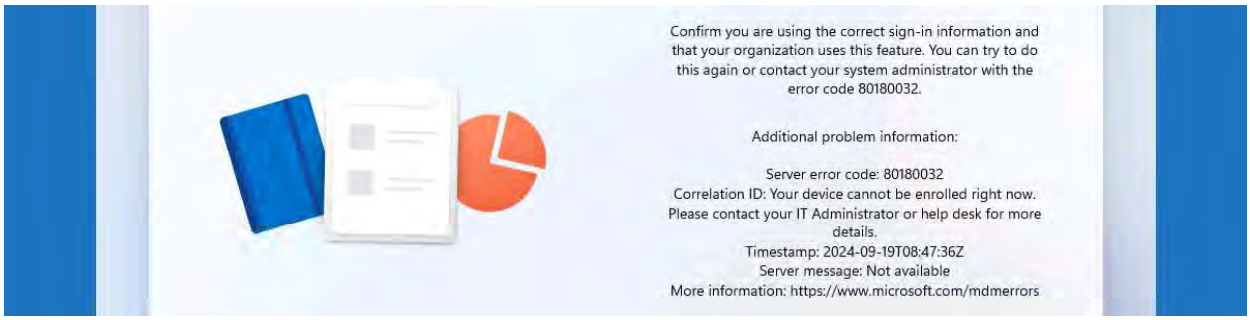
Instructions for Contributors

citrix

Site feedback | Your Privacy Choices | Privacy and legal terms | Cookie preferences | docs.cloud.com

© 1999-2025 Cloud Software Group, Inc. All rights reserved.

EXHIBIT 8



Getting started with Windows enrollment attestation

September 30, 2024by Peter van der Woude

This week is all about adding an additional layer of protection to the enrollment of Windows devices. That additional layer of protection is Windows enrollment attestation. Windows enrollment attestation is focused on making the process of enrolling into Microsoft Intune more secure and trustworthy for Windows devices. It relies on using the Trusted Platform Module (TPM) to store the private keys of the MDM certificate from Microsoft Intune and the access token from Microsoft Entra. That information is attested during the enrollment of Windows devices, making it less prone to tampering. That should provide better protection against attackers that for example steal an Intune MDM certificate. This blog post will start with a brief introduction about Windows enrollment attestation, followed with the central insights and the available remote actions.

> **Important**: Windows enrollment attestation is not supported for VMs, not even with vTPMs. Physical devices only.

Introducing Windows enrollment attestation

Privacy - Terms

When looking at the enrollment of Windows devices, the security of that enrollment process becomes more and more important. That's exactly where Windows enrollment attestation comes into play. Windows enrollment attestation adds an additional layer of security to the enrollment process of Windows devices, with the focus on making it more secure and trustworthy. That additional security and trustworthiness is achieved by relying on storing the private keys of the MDM certificate and the access token on the local TPM. That information is retrieved during the registration with Microsoft Entra, followed with the attest and enrollment with Microsoft Intune. Attesting the availability of that information in the TPM makes sure that only trusted Windows devices can perform the enrollment with Microsoft Intune. The great part is that the information can now also be actually used during the enrollment of Windows devices. A new property is available per device that can be used to create a filter. That new property is *device.IsTpmAttested* and that can be used to create a filter that can be used with enrollment restrictions.

> **Note**: Windows enrollment attestation can be used to prevent enrollment of Windows devices that are not attested.

Verifying Windows device attestation status

The good thing is that the device attestation status is now also available in a report. That report provides an overview of the attestation of all Windows devices within the environment and is available in the **Microsoft Intune admin center** via **Reports** > **Device management** > **Device attestation status**. An example is provided below in Figure 1. That example is filtered to show a few of the most important and most impactful properties, being *Device Model*, *OS Version* and *Device Attestation Status*. The latter is of course pretty obvious as it indicates the status of the device attestation. The other two properties are important as only physical devices are supported and only Windows versions after a specific version.

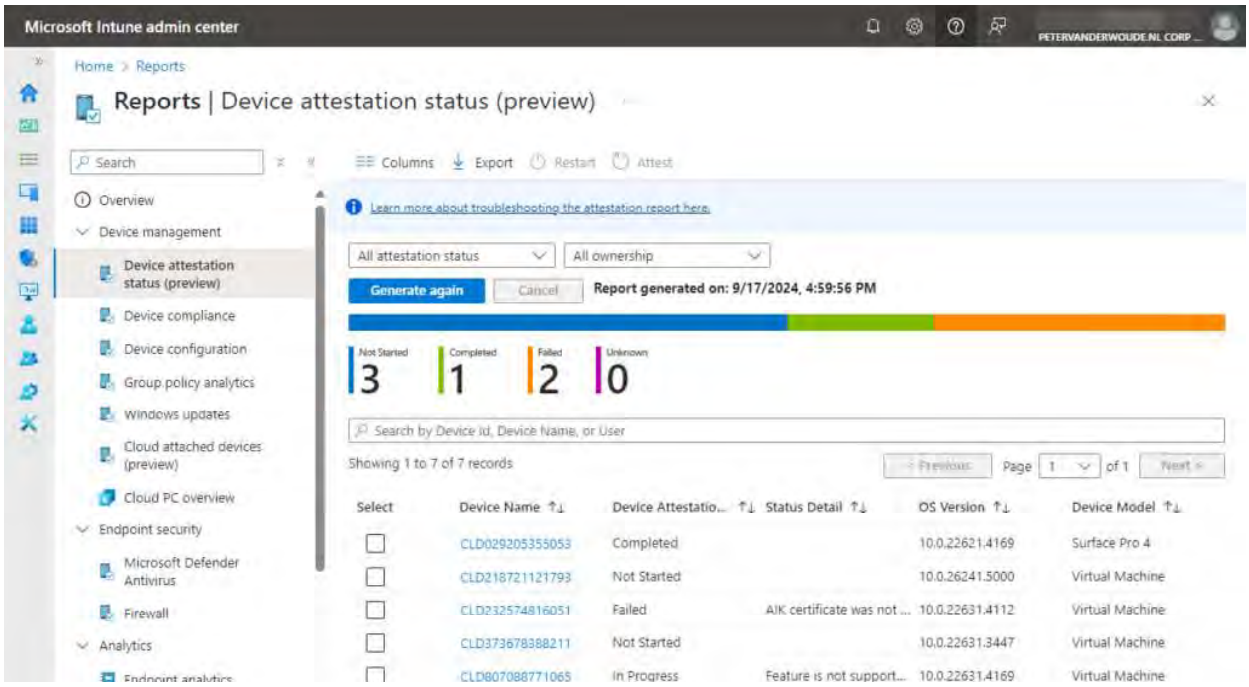


Figure 1: Overview of the device attestation status

Note: Of course the availability of TPM 2.0 is also important, but that's pretty standard nowadays.

Attesting Windows devices remotely

When devices are showing as not successfully attested, for whatever reason, a remote action can be triggered to attest the device. That action can be triggered via the *Device attestation report*, as shown below in Figure 2, by simply selecting the device and clicking on **Attest**. That will trigger a remote action similar to any remote action for a device. It will also be visible within the *Device actions status* overview of the device, as the *initiateMobileDeviceManagementKeyRecovery* action. That action will initiate an MDM key recovery and TPM attestation that basically locally on the device triggers the *InitiateRecovery* action of the DMClient CSP (and a bit more).

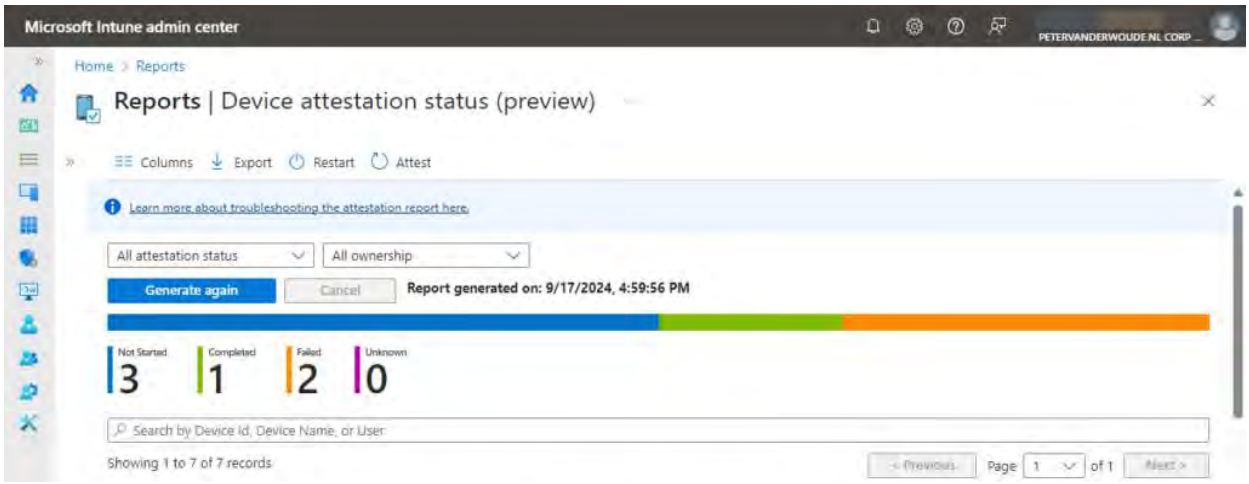


Figure 2: Overview of the remote attest action

Filtering Windows devices based on the attestation

The best part of Windows device attestation is the enrollment part, making it Windows enrollment attestation. That provides us with the ability to verify before the enrollment of a device, if that device is attested. For that, the new property *IsTpmAttested* is available. That property is not yet available, but can be configured via the rule syntax. That filter rule can then be used to configure specific enrollment restrictions. The following steps walk through the process of creating a filter rule for devices that are not attested.

1. Open the Microsoft Intune admin center portal navigate to **Devices** > **Organize devices** > **Filters**
2. On the **Devices | Filters**page, click **Create** > **Managed devices**
3. On the **Basics** page, specify a unique name for the filter, select **Windows 10 and later** as platform and click **Next**
4. On the **Rules** page, as shown below in Figure 3, edit the rule syntax, specify *device.IsTpmAttested -eq "False"* and click **Next**

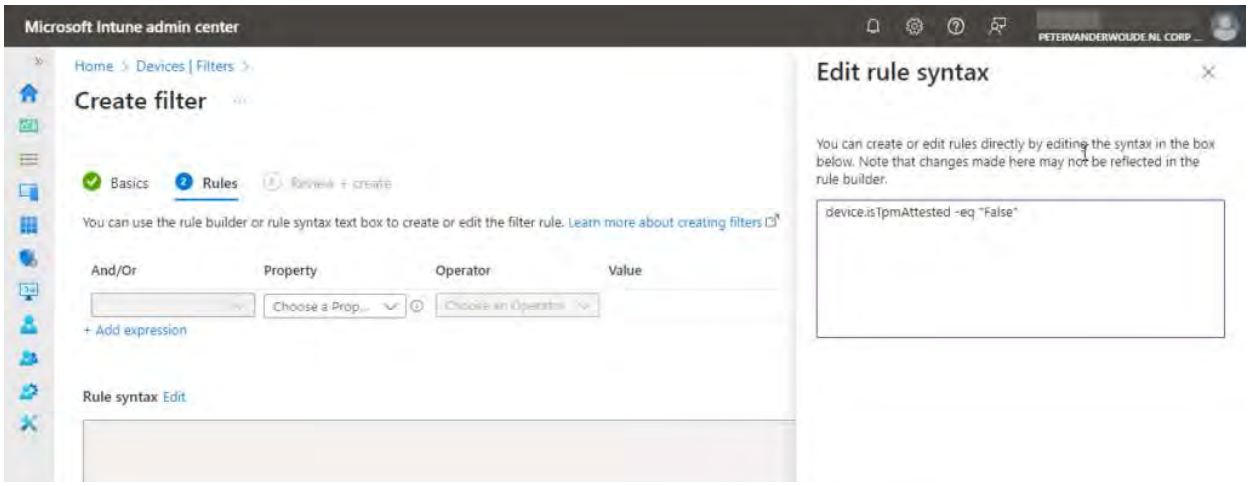


Figure 3: Overview of the creation of a filter based on the attestation status

5. On the **Review + create** page, verify the configuration and click **Create**

After the creation of the filter, it can be used to block the enrollment of the devices that match that filter. In other words, devices that have not successfully attested. That can be achieved by using a *Device enrollment restriction* When a device is being enrolled that is not successfully attested, an error will show similar to the header image.

More information

For more information about Windows enrollment attestation, refer to the following docs.

- Windows enrollment attestation | Microsoft Learn
- DMClient CSP | Microsoft Learn
- initiateMobileDeviceManagementKeyRecovery action – Microsoft Graph beta | Microsoft Learn
- Boost security with Microsoft Intune device attestation | Microsoft Intune blog

---

Discover more from All about Microsoft Intune

Subscribe to get the latest posts sent to your email.

Type your email... SUBSCRIBE

Microsoft Intune, Windows 10, Windows 11, Windows AutoPilot, Windows enrollment attestation

Microsoft Intune, Windows 10, Windows 11, Windows AutoPilot, Windows enrollment attestation

‹ Understanding the local diagnosing and troubleshooting options for Endpoint Privilege Management

› Easily managing Personal Data Encryption for known Windows folders

18 thoughts on “Getting started with Windows enrollment attestation”

Thomas B
October 1, 2024 at 08:24

Thanks for the clear write-up Peter!

Reply

Thomas B
October 1, 2024 at 09:00

One part I wasn’t able to distill is how the certificates are initially provisioned to the TPM – e.g. are they doing something nice like ACME with DA or is a cert generated on the infra side and then somehow either pre-installed or provisioned during autopilot enrolment – have you found that?

Reply

K.Mizra LLC, Ex. 2001, p. 110
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468



Peter van der Woude
October 7, 2024 at 20:23

I would say the latter. It's, however, not really clear documented. I know that Rudy looked a little bit under the hood and wrote this story:
https://call4cloud.nl/2024/07/istpmattested-enrollment-attestation/
Regards, Peter

Reply



Thomas B
October 8, 2024 at 10:09

Thank you! That does seem to confirm that the cert is initially created outside of the TPM.

Reply

Pingback: Microsoft Roadmap, messagecenter en blogs updates van 17-10-2024 - KbWorks - SharePoint & Teams Specialist

Pingback: Microsoft Roadmap, messagecenter en blogs updates van 20-10-2024 - KbWorks - SharePoint & Teams Specialist

Pingback: Microsoft Roadmap, messagecenter en blogs updates van 21-10-2024 - KbWorks - SharePoint & Teams Specialist

K.Mizra LLC, Ex. 2001, p. 111
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

Pingback: Microsoft Roadmap, messagecenter en blogs updates van 22-10-2024 - KbWorks - SharePoint & Teams Specialist

Pingback: Microsoft Roadmap, messagecenter en blogs updates van 23-10-2024 - KbWorks - SharePoint & Teams Specialist

Pingback: Microsoft Roadmap, messagecenter en blogs updates van 24-10-2024 - KbWorks - SharePoint & Teams Specialist

Pingback: Microsoft Roadmap, messagecenter en blogs updates van 25-10-2024 - KbWorks - SharePoint & Teams Specialist

Pingback: Microsoft Roadmap, messagecenter en blogs updates van 26-10-2024 - KbWorks - SharePoint & Teams Specialist

Pingback: Microsoft Roadmap, messagecenter en blogs updates van 27-10-2024 - KbWorks - SharePoint & Teams Specialist

Pingback: Microsoft Roadmap, messagecenter en blogs updates van 28-10-2024 - KbWorks - SharePoint & Teams Specialist



Mathieu Ait Azzouzene
November 19, 2024 at 15:52

It is pretty unclear if Windows Enrollment Attestation occurs during a User Driven Autopilot provisioning or if it is only triggered during self-deploying and pre-provisioning.

Reply

K.Mizra LLC, Ex. 2001, p. 112
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468



Peter van der Woude
November 25, 2024 at 20:14

It's for all physical devices with TPM2.0.
Regards, Peter

Reply

Mathieu Ait Azzouzene
November 25, 2024 at 20:17

My question is not about the type of TPM, but about when this attestation occur. Will a user will be allowed to perform a user driven Autopilot if this enrollment restriction is in place? I mean will the TPM will be attested before Intune enrollment kicks in during a user driven Autopilot provisioning?

Reply



Peter van der Woude
November 25, 2024 at 20:37

It's also for user driven deployments. That's why I mentioned "all devices".
Regards, Peter

Reply

Leave a Comment

Name *

Email *

Website

Notify me of follow-up comments by email.

Notify me of new posts by email.

POST COMMENT

This site uses Akismet to reduce spam. Learn how your comment data is processed.

AWARD



SUBSCRIBE

Type your email...

SUBSCRIBE

Join 1,304 other subscribers

SUPPORT



MORE SUPPORT

If you enjoy reading my content, please consider buying me a coffee. That helps me with staying awake, and keeping my blog online.

Buy me a coffee!

ABOUT ME

I'm Peter van der Woude, born in 1983 and I'm living together with my wife and two sons in the Netherlands. I work as a Principal Consultant at InSpark and my main focus is helping customers in their road to a modern workplace (using Microsoft Intune).

CALENDAR

April 2025

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			



« Mar

LATEST POSTS

Notifying about Intune audit logs with Azure Logic Apps and notifications in Teams
by Peter van der Woude
April 14, 2025

Easier managing account management modes for Windows LAPS
by Peter van der Woude
April 7, 2025

Getting started with the Global Secure Access client for Android
by Peter van der Woude
March 31, 2025

Using a QR code to easily sign in to apps on shared Android devices
by Peter van der Woude
March 17, 2025

Easier managing Device Control in Microsoft Defender for Endpoint using Microsoft Intune
by Peter van der Woude
March 10, 2025

TAGS

ANDROID | ANDROID ENTERPRISE | AZURE AD | CONDITIONAL ACCESS | CONFIGMGR | CONFIGMGR 2007 | CONFIGMGR 2012 | IOS | IPADOS | MDM | MICROSOFT ENDPOINT MANAGER | MICROSOFT INTUNE | OMA-DM | OMA-URI | POWERSHELL | SCCM | TASK SEQUENCE | WINDOWS 10 | WINDOWS 11 | WINDOWS AUTOPILOT

© 2009 - 2025 PETER VAN DER WOUDE

K.Mizra LLC, Ex. 2001, p. 117
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

EXHIBIT 9

citrix

Device Posture

Ex. 9, p. 1
K.Mizra LLC, Ex. 2001, p. 119
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

Contents

Device Posture overview 2

What's new 7

Configure Device Posture policies 11

Configure Device Posture global settings 18

Third-party integration with device posture 27

End user flow 39

Device Posture service in test mode 42

Export of Device Posture user events into SIEM systems 45

Troubleshoot Device Posture issues using DaaS Monitor 46

Device Posture logs and events 47

Manage Citrix Endpoint Analysis client for Device Posture service 49

Data Governance 52

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

Device Posture overview

March 3, 2025

Citrix Device Posture service is a cloud-based solution that helps admins to enforce certain requirements that the end devices must meet to gain access to Citrix DaaS (virtual apps and desktops) or Citrix Secure Private Access resources (SaaS, Web apps, TCP, and UDP apps). Establishing device trust by checking the device's posture is critical to implement zero-trust-based access. Device Posture service enforces zero trust principles in your network by checking the end devices for compliance (managed/BYOD and security posture) before allowing an end user to log in.

Prerequisites

- Licensing requirements: The Device Posture service is available as part of the Universal Hybrid Multi Cloud License (UHMC) and Platform License (CPL). For more information, see https://www.citrix.com/buy/licensing/product.html.
- Supported platforms:
  - Windows (10 and 11)
  - macOS 13 Ventura
  - macOS 12 Monterey
  - iOS
  - IGEL
  - Linux

> **Note:**
>
> - A device running on a non-supported platform is marked as non-compliant by default. You can change the classification from **Non-compliant** to **Denied login** from the **Settings** tab on the Device Posture page. For the definitions of "compliant"and "non-compliant,"see Definitions.
> - A device that is running on a supported platform but does not match any pre-defined device posture policy is marked as non-compliant, by default. You can change the classification from **Non-compliant** to **Denied login** from the **Settings** tab on the Device Posture page.
> - For iOS support on Device Posture service, the EPA client is built in as part of the Citrix Workspace app for iOS. For details on the versions, see Citrix Workspace app for iOS.
> - For IGEL OS support on the Device Posture service, the EPA client is built in as part of the IGEL OS. Contact the IGEL support team for installing the EPA client on the IGEL

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

> devices.

- Citrix Device Posture client (EPA client): A lightweight application that must be installed on the endpoint device to run device posture scans. This application does not require local admin rights to download and install on an endpoint.

  > **Note:**
  >
  > If you're using a device certificate check, then you must install the EPA client with administrative rights.

- Supported browsers: Chrome, Edge, and Firefox.
- Firewall configuration: To allow the Device Posture service to update the EPA clients on an end device, the firewall/proxy must be configured to allow the following domains:
  - `https://swa-ui-cdn-endpoint-prod.azureedge.net`
  - `https://productioniconstorage.blob.core.windows.net`
  - *.netscalergateway.net
  - *.nssvc.net
  - *.cloud.com
  - *.pendo.io
  - *.citrixworkspacesapi.net

How it works

The admins can create device posture policies to check the posture of endpoint devices and determine whether an endpoint device is allowed or denied login. The devices, which are allowed to log in are further classified as compliant or non-compliant. Users can log in from a browser or the Citrix Workspace app.

Following are the high-level conditions used to classify a device as compliant, non-compliant, and denied login.

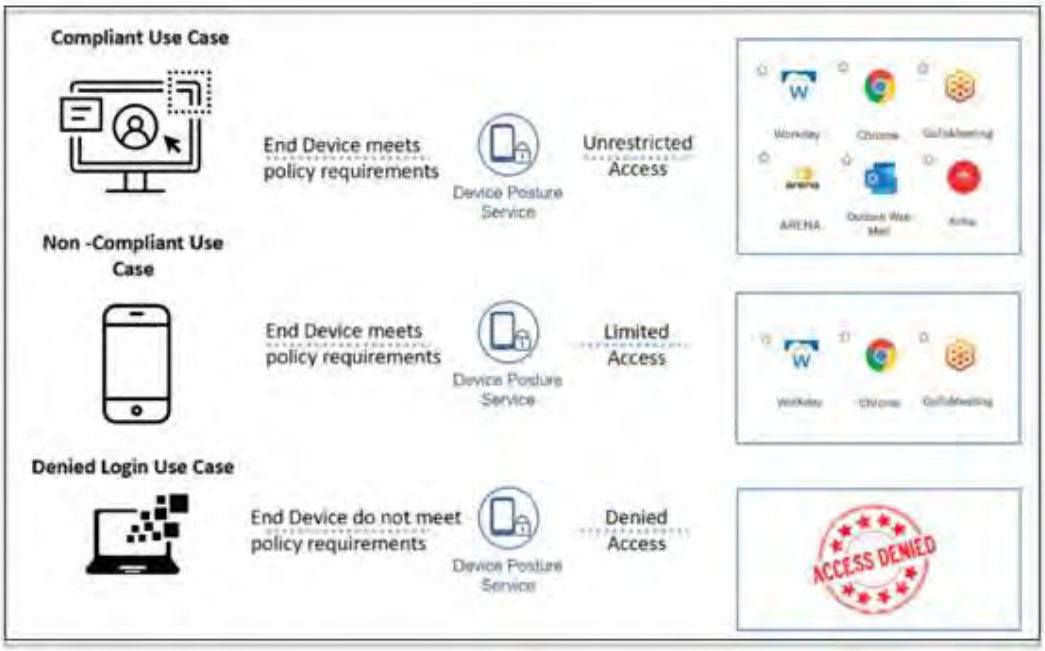
- **Compliant devices** –A device that meets the pre-configured policy requirements and is allowed to log in into the company's network with full or unrestricted access to Citrix Secure Private Access resources or Citrix DaaS resources.
- **Non-Compliant devices** - A device that meets the pre-configured policy requirements and is allowed to log in into the company's network with partial or restricted access to Citrix Secure Private Access resources or Citrix DaaS resources based on your DaaS and Secure Private Access configuration for Device Posture.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

  - DaaS: Administrators can specify the delivery groups to restrict access for non-compliant devices. For details on how to configure restricted access to delivery groups, see Citrix DaaS configuration with Device Posture.
  - Secure Private Access: Administrators can choose specific applications to restrict access for these devices. For details on how to restrict access to specific applications, see Citrix Secure Private Access configuration with Device Posture.

- **Denied login: -** A device that fails to meet the policy requirements is denied login.

The classification of devices as **compliant**, **non-compliant**, and **denied login** is passed onto the Citrix DaaS and Citrix Secure Private Access service that in turn uses the device classification to provide smart access capabilities.



Note:

- The device posture policies must be configured specifically for each platform. For example, for macOS, an admin can allow access for the devices that have a specific OS version. Similarly, for Windows, the admin can configure policies to include a specific authorization file, registry settings, and so on.
- Device posture scans are done only during pre-authentication/before logging in.
- For definitions of "compliant"and "non-compliant,"see Definitions.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

Scans supported by the Device Posture service

The following scans are supported by the Citrix Device Posture service:

Windows	macOS	iOS	IGEL	Linux
Citrix Workspace App Version	Citrix Workspace App Version	Citrix Workspace App Version	-	-
OS Version	OS Version	OS Version	-	OS Version
File (Exists, Last Modified Time, MD5)	File (Exists, Last Modified Time, MD5)	-	File (Exists, Last Modified Time, MD5)	File (Exists, Last Modified Time, MD5)
Geo Location	Geo Location	-	-	-
Network Location	Network Location	-	-	-
MAC Address	MAC Address	-	-	-
Process (Exists, MD5, SHA1)	Process (Exists, MD5, SHA1)	-	-	Process (Exists, MD5, SHA1)
Microsoft Endpoint Manager	Microsoft Endpoint Manager	-	-	-
CrowdStrike	CrowdStrike	-	-	-
Device Certificate	Device Certificate	-	-	-
Web Browser	Web Browser	-	-	-
Antivirus	Antivirus	-	-	-
Non-Numeric Registry (32 Bit)	-	-	-	-
Non-Numeric Registry (64 Bit)	-	-	-	-
Numeric Registry (32 Bit)	-	-	-	-
Numeric Registry (64 Bit)	-	-	-	-
Windows Update Installation Type	-	-	-	-
Windows Update Last Update Check	-	-	-	-

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

Windows	macOS	iOS	IGEL	Linux
External Device	-	-	-	-
Connected Network Settings	-	-	-	-
-	-	-	-	Mount Point

Note:

- Starting from EPA library version 24.9.1.1, the **Windows Update Last Update** scan also includes updates installed through BigFix, Microsoft Intune, and other third-party tools, in addition to those installed through the Windows Auto Upgrade service.
- The **Windows Update Last Update** scan is not applicable for manually installed updates where the update patch is downloaded and installed by the user.
- For iOS support on Device Posture service, the EPA client is built in as part of the Citrix Workspace app for iOS. For details on the versions, see Citrix Workspace app for iOS.

Third-party integration with the Device Posture service

In addition to the native scans offered by the Device Posture service, the service can also be integrated with the following third-party solutions on Windows and macOS.

- Microsoft Intune. For details, see Microsoft Intune integration with Device Posture.
- CrowdStrike. For details, see CrowdStrike integration with Device Posture.

Known limitations

- The time taken for the device posture functionality to be enabled or disabled after the device posture toggle button is turned On or Off can take a few minutes to an hour.
- Any changes in the device posture configuration do not take effect immediately. It might take around 10 minutes for the changes to take effect.
- If you have enabled the Service Continuity option in Citrix Workspace and if the Device Posture service is down, users might be unable to sign in to Workspace. This is because Citrix Workspace enumerates apps and desktops based on local cache on the user device.
- If you have configured a long-lived token and password on Citrix Workspace, the device posture scan does not work for this configuration. The devices are scanned only when the users log in to Citrix Workspace.
- Each platform can have a maximum of 10 policies and each policy can have a maximum of 10 rules.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

- Role-based access is not supported with the Device Posture service.
- The Device Posture service is not supported in Citrix Cloud Japan.

Quality of service

- Performance: Under ideal conditions, the Device Posture service adds an additional 2 seconds of delay during login. This delay might increase depending on additional configurations such as third-party integrations like Microsoft Intune.
- Resiliency: Device Posture service is highly resilient with multiple POPs to ensure that there's no downtime.

Definitions

The terms "compliant"and "non-compliant"in reference to the Device Posture service are defined as follows.

- **Compliant devices**–A device that meets the pre-configured policy requirements and is allowed to log into the company's network with full or unrestricted access to Citrix Secure Private Access resources or Citrix DaaS resources.
- **Non-Compliant devices** - A device that meets the pre-configured policy requirements and is allowed to log in into the company's network with partial or restricted access to Citrix Secure Private Access resources or Citrix DaaS resources.

What's new

February 24, 2025

21 Feb 2025

- **Device compliance with event-driven posture checks**

  Starting with EPA Library version 24.11.1.1, Device Posture introduces enhanced checks to assess the compliance of your devices.

  The following two scans are added to provide more control over device compliance:

  - **External Device Connected:** This scan detects if an external USB storage device is connected to the endpoint.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

  - **Network Settings:** This scan determines if the device is connected to a protected Wi-Fi network.

When any of the following events occur, Device Posture re-evaluates the device's compliance status:

- The **External Device Connected** scan is configured, and an external USB storage device is inserted or removed.
- The **Network Settings** scan is configured, and a switch between an open network and a protected network is detected.
- The **Windows Defender Firewall** scan is configured, and it's enabled or disabled.

Access to the resources is updated in accordance with the new compliance status. This ensures that access to resources is granted or revoked based on the device's real-time compliance status.

For details, see Device compliance with event-driven posture checks.

> **Note:**
>
> This feature is supported only on the Windows platform.

- **TCP/UDP active sessions terminated after a downgrade**

  For TCP/UDP applications launched from the Citrix Secure Access client, if a device posture scan results in a downgrade (for example from **Compliant** to **Non-compliant** or **Denied access**), users receive a notification, and after 5 minutes, the active sessions are terminated. For details, see Periodic scanning of devices.

  > **Note:**
  >
  > This feature is supported only on the Windows platform and is available from the End Point Analysis (EPA) client version 24.8.1.19 and library version 24.9.1.1.

- **Enhancements to the Windows Update Last Update scan**

  Starting from EPA library version 24.9.1.1, the **Windows Update Last Update** scan also includes updates installed through BigFix, Microsoft Intune, and other third-party tools, in addition to those installed through the Windows Auto Upgrade service.

04 Dec 2024

- **Jamf Pro integration with Device Posture**

  In addition to the native scans offered by the Device Posture service, the Device Posture service is also integrated with Jamf Pro. For details, see Jamf Pro integration with Device Posture.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

29 May 2024

- **Availability of Device Posture service in test mode**

  The Device Posture service is also available in test mode wherein admins can test the Device Posture service before enabling it on their production environment. This enables the admins to analyze the impact of the device posture scans on the end user devices and then plan their course of action accordingly before enabling it on production. For details, see Device Posture service in test mode.

- **Periodic scanning of devices

  You can now enable periodic scanning of Windows devices for the configured checks every 30 minutes. For details, see Periodic scanning of devices.

14 May 2024

- **Skip device posture checks**

  Admins can allow the end users to skip the device posture checks on their devices. For details, see Skip device posture checks.

- **Device posture dashboard**

  The Device Posture service portal now has a dashboard for monitoring and troubleshooting logs. Admins can now use this dashboard for monitoring and troubleshooting purposes. For details, see Device posture logs.

- **General availability of browser and antivirus checks**

  The browser and antivirus checks are now generally available. For details, see Scans supported by device posture.

- **General availability of custom messages**

  The option to add customized messages when access is denied is now generally available. For details, see Customized messages for access denied scenarios.

26 March 2024

- **Custom workspace URLs support**

  Custom workspace URLs are now supported with the Device Posture service. You can use a URL that you own in addition to your cloud.com URL to access workspace. Ensure that you allow access to citrix.com from your network. For details on custom domains, see Configure a custom domain.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

12 February 2024

- **Support for browser and antivirus checks - Preview**

  Device Posture service now supports browser and antivirus checks. For details, see Scans supported by device posture.

23 January 2024

- **General availability of device certificate check with Device Posture service**

  Device certificate check with the Device Posture service is now generally available. For details, see Device certificate check.

- **Device Posture service preview features**

  Device Posture service now supports the following checks:

  - Device Posture service is now supported on the IGEL platforms.
  - Device Posture service now supports geolocation and network location checks.

  For details, see Device Posture.

11 September 2023

- **General availability of Device Posture Integration with Microsoft Intune**

  Device Posture Integration with Microsoft Intune is now generally available. For details, see Microsoft Intune integration with Device Posture.

30 August 2023

- **Manage Citrix Endpoint Analysis Client for Device Posture service**

  The EPA client can be used together with NetScaler and Device Posture. Some configuration changes are required to manage EPA client when used with NetScaler and Device Posture. For details, see Manage Citrix Endpoint Analysis Client for Device Posture service.

28 August 2023

- **Device Posture service support on iOS platforms - Preview**

  Device Posture service is now supported on iOS platforms. For details, see Device Posture.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

22 August 2023

- **Device Certificate check with Citrix Device Posture service - Preview**

  Citrix Device Posture service can now enable contextual access (Smart Access) to Citrix DaaS and Secure Private Access resources by checking the end device's certificate against a corporate certificate authority to determine if the end device can be trusted. For details, see Device certificate check.

17 August 2023

- **Device Posture events on Citrix DaaS Monitor**

  Device Posture service events and monitoring logs are now searchable on DaaS Monitor. For details, see Device posture events on Citrix DaaS Monitor.

23 January 2023

- **Device posture service**

  Citrix Device Posture service is a cloud-based solution that helps admins to enforce certain requirements that the end devices must meet to gain access to Citrix DaaS (virtual apps and desktops) or Citrix Secure Private Access resources (SaaS, Web apps, TCP, and UDP apps). For details, see Device Posture.

  [AAUTH-90]

- **Microsoft Endpoint Manager integration with Device Posture**

  In addition to the native scans offered by the Device Posture service, the Device Posture service can also be integrated with other third-party solutions. Device Posture is integrated with Microsoft Endpoint Manager (MEM) on Windows and macOS. For details, see Microsoft Endpoint Manager integration with Device Posture.

  [ACS-1399]

Configure Device Posture policies

October 8, 2024

Device posture is a combination of policies and rules that a device must meet to gain access to the resources. Each policy is attached with one of the actions namely compliant, non-compliant, and

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

denied login. In addition, each policy is associated with a priority and the policy evaluation stops if a policy evaluates to true and the associated action is taken.

1. Sign in to Citrix Cloud, and then select **Identity and Access Management** from the hamburger menu.
2. Click the **Device Posture,** tab and then click **Manage**.

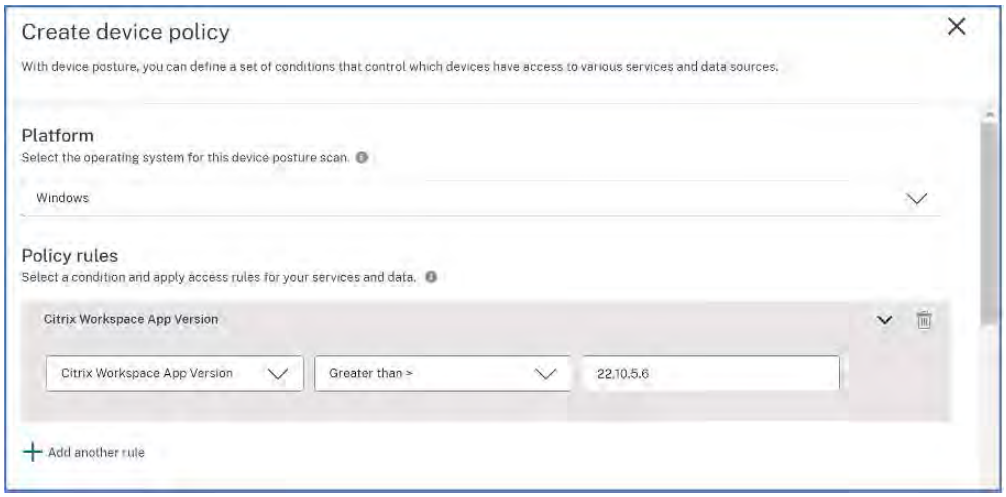
   > **Note:**
   >
   > - Secure Private Access service customers can directly click **Device Posture** on the left navigation in the admin user interface.
   > - For the first-time users, the Device Posture landing page prompts you to create a device posture policy. Device posture policy must individually be configured for each platform. Once you create a device posture policy, it gets listed under the appropriate platforms.
   > - A policy comes into effect only after device posture is enabled. To enable device posture, slide the **Device posture is disabled** toggle on the right hand top corner to **ON**.

3. Click **Create device policy**.
4. In **Platform**, select the platform for which you want to apply a policy. You can change the platform from Windows to macOS or conversely irrespective of the tab that you selected on the Device Posture home page.
5. In **Policy rules**, select the check that you want to perform as part of device posture and select the conditions that must be matched.

   > **Note:**
   >
   > - For device certificate check, ensure that the issuer certificate exists on the device. Else, you can import a device certificate while creating the device posture policy or upload the certificate from **Settings** on the Device Posture home page. For details, Device certificate check.
   > - For the device certificate check, the EPA client on the end device must be installed with administrative rights.
   > - The device certificate check with the Device Posture service does not support the Certificate Revocation check.

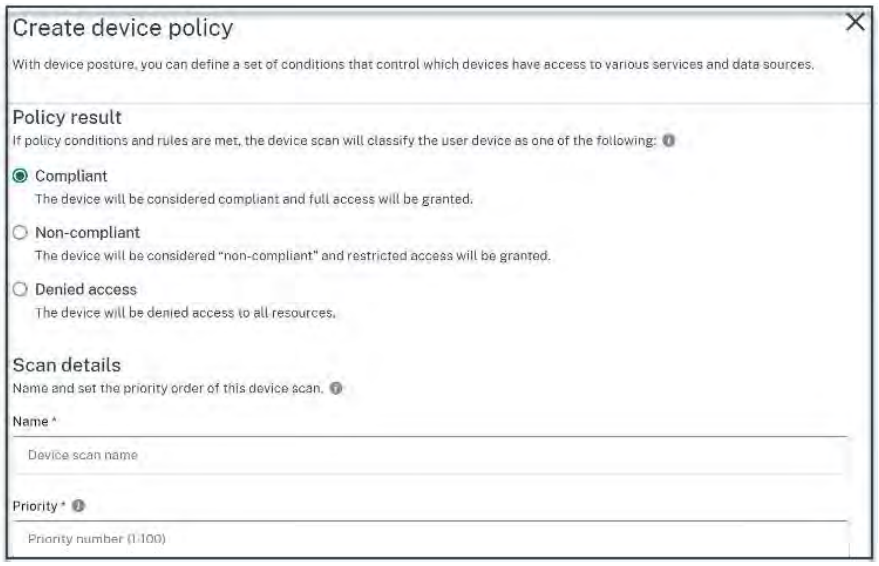
6. Click **Add another rule** to create multiple rules. An AND condition is applied on multiple rules.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.



7. In **Policy result** based on the conditions that you've configured, select the type under which the device scan must classify the user device.
    - Compliant
    - Non-compliant
    - Denied access
8. Enter a name for the policy.
9. In **Priority**, enter the order in which the policies must be evaluated.
    - You can enter a value between 1 through 100. It's recommended that you configure deny policies with higher priority, followed by non-compliant, and finally compliant.
    - The priority with the lower value has the highest preference.
    - Only the policies that are enabled are evaluated based on the priority.
10. Click **Create**.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.



> **Important:**
>
> You must turn the **Enable when created** toggle switch to **ON** for the device posture policies to take effect. Before you enable the policies, it's recommended that you ensure that the policies are correctly configured and you're performing these tasks in your test setup.

Configure contextual access (smart access) using device posture

After the device posture verification, the device is allowed to log in and classified as compliant or non-compliant. This information is available as tags to the Citrix DaaS service and Citrix Secure Private Access service and is used to provide contextual access based on device posture. Therefore, Citrix DaaS and Citrix Secure Private Access must be configured to enforce access control using device posture tags.

Citrix DaaS configuration with Device Posture

To restrict access to the DaaS resources, you can specify the delivery groups on which access restrictions must be enforced and configure the access control restrictions using the Device Posture tags.

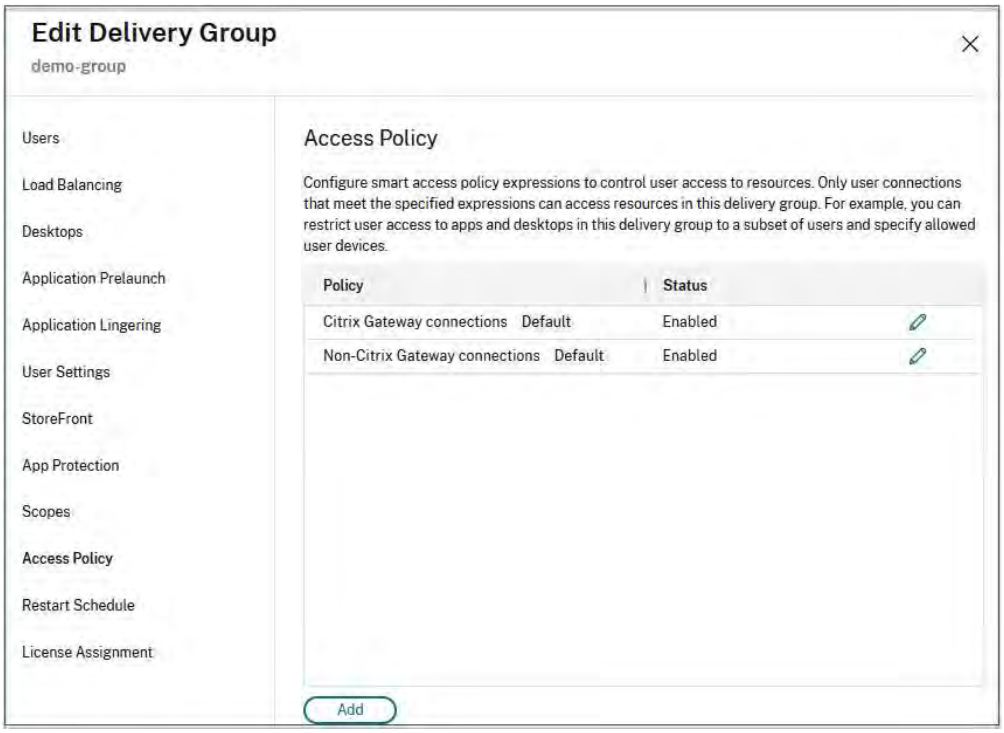
Prerequisites:

Ensure that the Adaptive Access feature is enabled (**Citrix Workspace > Access > Adaptive Access**). For details, see Enable the Adaptive Access feature.

1. Sign into Citrix Cloud.
2. On the **DaaS** tile, click **Manage**.

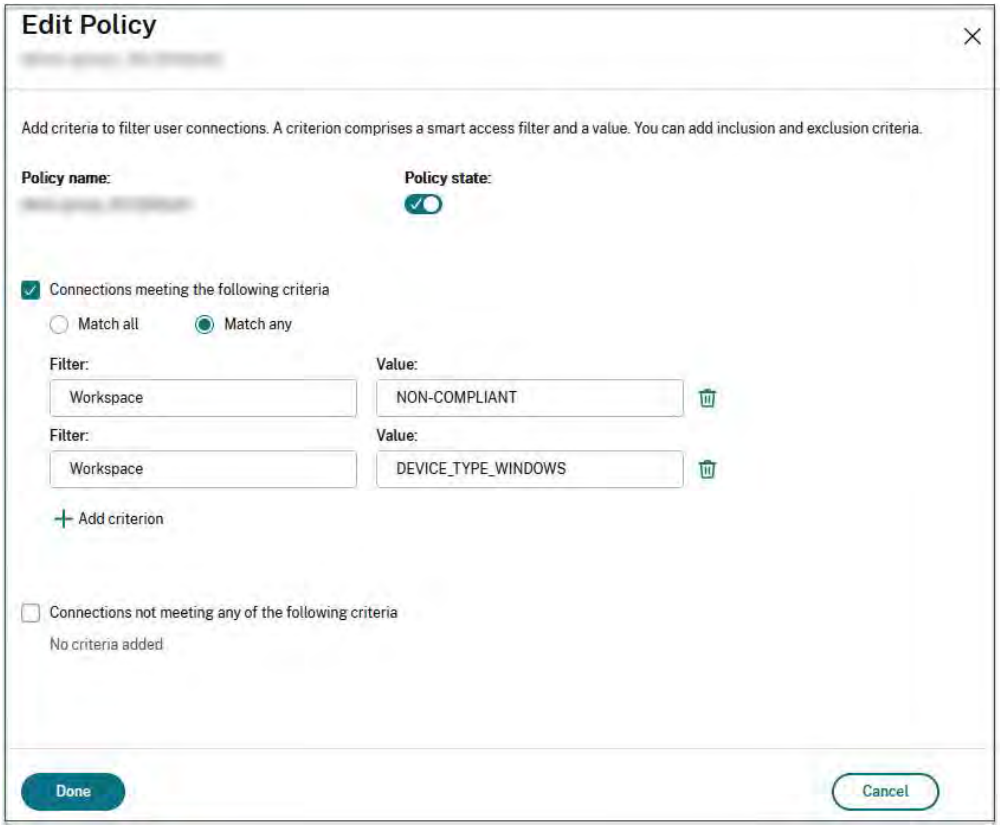
© 1999–2025 Cloud Software Group, Inc. All rights reserved.

3. Go to the **Delivery Group** section from the left-hand menu.
4. Select the delivery group for which you want to configure access control based on device posture and click **Edit**.
5. In the **Edit Delivery Group** page, click **Access Policy**.
6. Click the edit icon on the **Citrix Gateway connections** row to edit the gateway connections policy.



a) On the Edit policy page, select **Connections meeting the following criteria.**
b) Select **Match any**, and then click **Add criterion**.
c) Add criteria for all location tags you configured in Configure network locations: Type **Workspace** for **Filter** and **COMPLIANT** or **NON-COMPLIANT** for **Value**.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.



Note:

The syntax for the device classification tags must be entered exactly as captured earlier, that is all in uppercase (**COMPLIANT** and **NON-COMPLIANT**). Else the device posture policies do not work as intended.

In addition to the device classification tags, the Device Posture service also returns the operating system tag and the access policy tag associated with the device. The operating system tags and the access policy tags must be entered in uppercase only.

- DEVICE_TYPE_WINDOWS
- DEVICE_TYPE_MAC
- Exact policy name (uppercase)

Citrix Secure Private Access configuration with Device Posture

To restrict access to the Secure Private Access resources, you can select the applications on which access restrictions must be enforced and configure the access control restrictions using the Device Posture tags.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

1. Sign into Citrix Cloud.
2. On the Secure Private Access tile, click **Manage**.
3. Click **Access Policies** on the left navigation and then click **Create policy**.
4. Enter the policy name and description of the policy.
5. In **Applications**, select the app or set of apps on which this policy must be enforced.
6. Click **Create Rule** to create rules for the policy.
7. Enter the rule name and a brief description of the rule, and then click **Next**.
8. Select the users'conditions. The **Users** condition is a mandatory condition to be met to grant access to the applications for the users.
9. Click **+** to add device posture condition.
10. Select **Device posture check** and the logical expression from the drop-down menu.
11. Enter one of the following values in the custom tags:
    - **Compliant** - For compliant devices
    - **Non-Compliant -** For non-compliant devices

    > **Note:**
    >
    > The tags must be entered exactly as captured earlier, using initial caps (**Compliant** and **Non-Compliant**). Otherwise, the device posture policies do not function as intended.

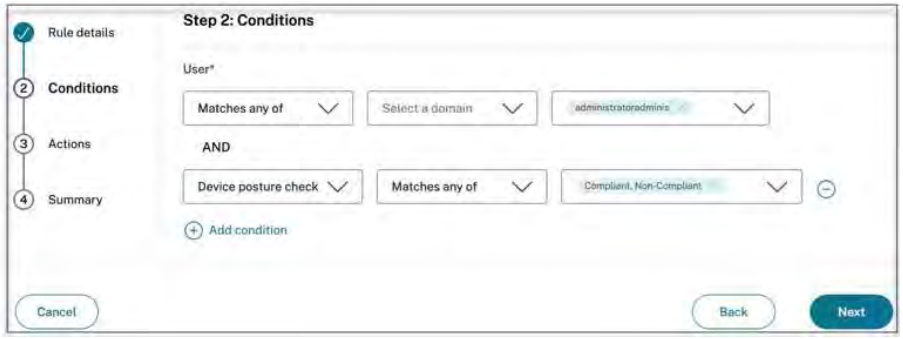
12. Click **Next**.
13. Select the actions that must be applied based on the condition evaluation, and then click **Next**.

    The Summary page displays the policy details.
14. You can verify the details and click **Finish**.

    For more details on creating access policies, see Configure an access policy with multiple rules.

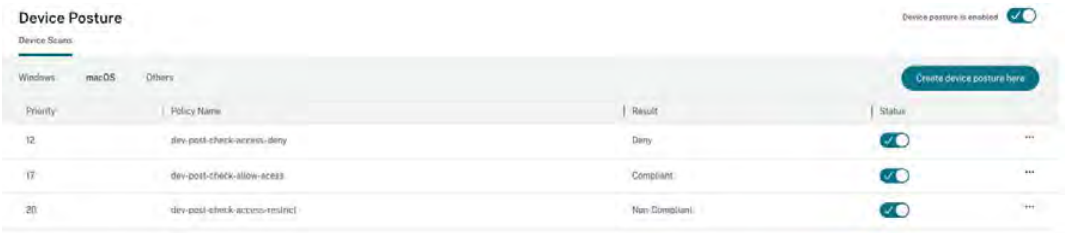
> **Note:**
>
> Any Secure Private Access application which isn't tagged as compliant or non-compliant in the access policy is treated as the default application and is accessible on all the endpoints regardless of device posture.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.



Edit a device posture policy

The configured device posture policies are listed under the specific platform in the **Device Scans** page. You can search for the policy you want to edit from this page. You can also enable, disable, or delete a policy from this page.



Configure Device Posture global settings

February 21, 2025

The following are some of the global Device Posture settings that you can configure as required.

- Device compliance with event-driven posture checks
- Periodic scanning of devices
- Customized messages for access denied scenarios
- Skip device posture checks
- Custom workspace URLs support
- Session Recording configuration with Device Posture
- Device certificate check

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

Device compliance with event-driven posture checks

Starting with EPA Library version 24.11.1.1, Device Posture introduces enhanced checks to assess the compliance of your devices.

The following two scans are added to provide more control over device compliance:

- **External Device Connected:** This scan detects if an external USB storage device is connected to the endpoint.
- **Network Settings:** This scan determines if the device is connected to a protected Wi-Fi network.

When any of the following events occur, Device Posture re-evaluates the device's compliance status:

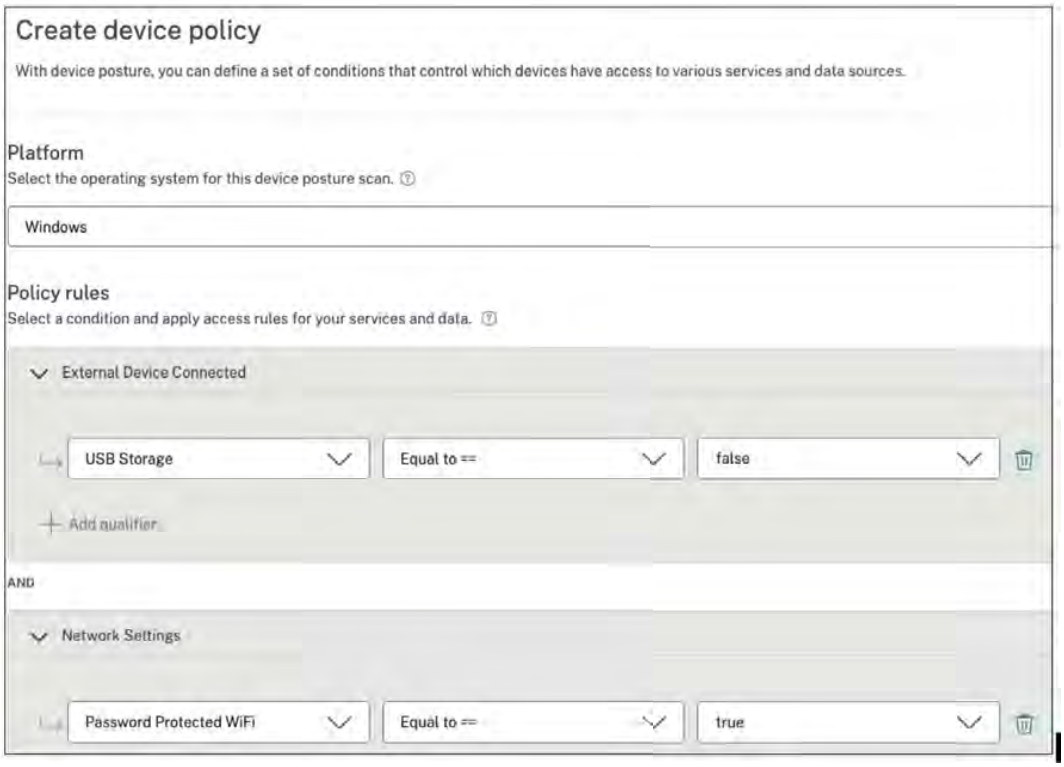
- The **External Device Connected** scan is configured, and an external USB storage device is inserted or removed.
- The **Network Settings** scan is configured, and a switch between an open network and a protected network is detected.
- The **Windows Defender Firewall** scan is configured, and it's enabled or disabled.

Access to the resources is updated in accordance with the new compliance status. This ensures that access to resources is granted or revoked based on the device's real-time compliance status.

> **Note:**
>
> Event-driven posture checks are currently supported only on Windows platforms. While both user and administrator installations can utilize the posture scans, real-time re-evaluation of device compliance based on events requires administrator privileges.

The following figure displays the **External Device Connected** and **Network Settings** scans configured in the Device Posture admin console.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.



Periodic scanning of devices

You can enable periodic scanning of Windows devices for the configured checks every 30 minutes. The EPA client on the end device must be installed with administrative rights to enable periodic scanning of devices. To enable periodic scanning, do the following:

1. Navigate to **Device Posture > Device Scans** and click **Settings**.
2. In **Periodic device posture scans** section, slide the toggle switch ON to enable periodic scanning of devices.



- For HTTP/HTTPS applications, if a device posture scan result changes from **Compliant** to **Non-compliant** or **Denied access**, access to the resources is updated in accordance with the new compliance status.
- For TCP/UDP applications launched from the Citrix Secure Access client, if a device posture scan results in a downgrade (for example from **Compliant** to **Non-compliant** or **Denied access**),

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

users receive a notification, and after 5 minutes, the active sessions are terminated.



Customized messages for access denied scenarios

Admins can customize the message that is displayed on the end device when access is denied.

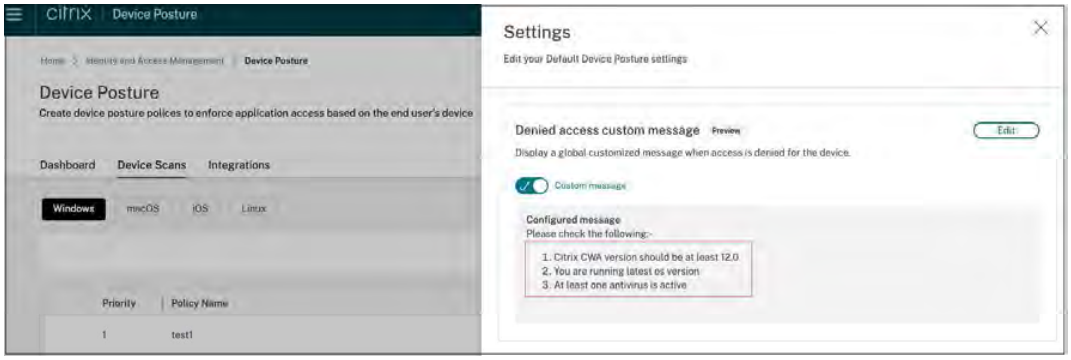
Perform the following steps to add customized messages:

1. Navigate to the **Device Posture > Device Scans** page.
2. Click **Settings**.
3. Click **Edit** and in the **Message** box, enter the message that must be displayed in access denied scenarios. You can enter a maximum of 256 characters.
4. Click **Enable custom message on save** to enforce the option of displaying the custom message. If you do not select this checkbox, the custom message is created but not displayed on the devices in access denied scenarios.

   Alternatively, you can enable the **Custom message** toggle switch on the **Settings** page to display the message on the devices.
5. Click **Save**.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

The following image displays a sample message added by the admin.



The following image displays the custom message that appears on the end user device when access is denied.



The message that you have entered appears whenever access is denied for the end device.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

Skip device posture checks

Admins can allow the end users to skip the device posture checks on their devices in the following scenarios:

- The device posture agent is not installed on the device.
- The EPA client installed on the device is not the latest version.

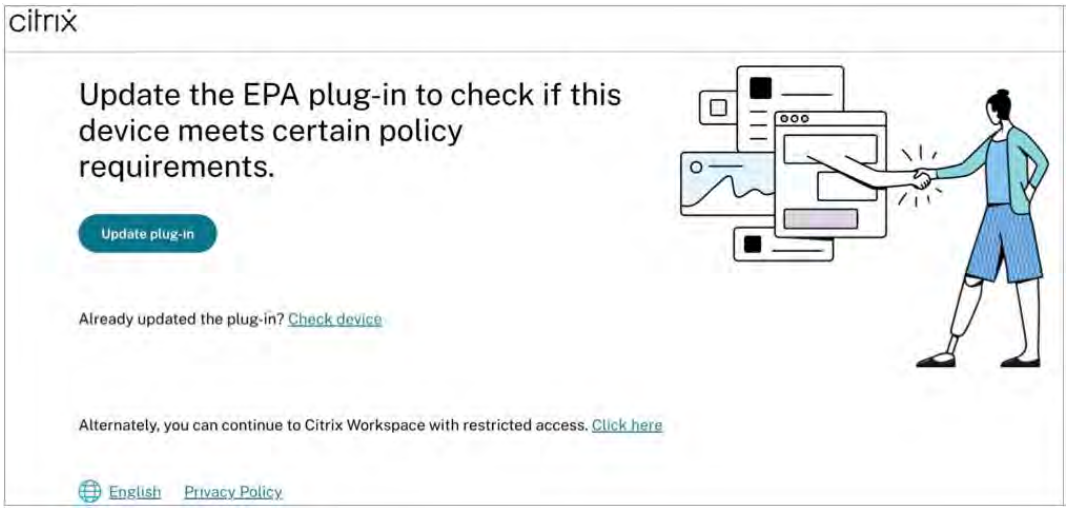
When the skip check feature is enabled, the default policy result (non-compliant) is enforced and the device is classified as non-compliant. End users are provided with partial or restricted access to the Citrix Secure Private Access or Citrix DaaS resources.

Enable skipping the device posture checks

1. Navigate to **Device Posture > Device Scans**.
2. Click **Settings**.
3. In the **Skip device posture check** section, slide the toggle switch ON to enable skipping of the device posture checks.

   When the **Skip device posture check** option is enabled and the end user logs on to the Citrix Workspace, the following message appears when the end user tries to download the client or upgrade the EPA version.

   Alternately, you can continue to Citrix Workspace with restricted access.



© 1999–2025 Cloud Software Group, Inc. All rights reserved.

Custom workspace URLs support

Custom workspace URLs are supported with the Device Posture service. You can use a URL that you own in addition to your cloud.com URL to access workspace. Ensure that you allow access to citrix.com from your network. For details on custom domains, see Configure a custom domain.

Session Recording configuration with Device Posture

Session Recording allows organizations to record on-screen user activity in virtual sessions. You can specify tags when creating a custom session recording policy, event detection policy, or event response policy. For an example, see Create a custom recording policy.

Device certificate check

To configure device certificate checks with the Device Posture service, admins must import an issuer certificate from their device. Once a valid issuer certificate is present in the Device Posture service, admins can use device certificate checks as part of device posture policies.

> **Points to note:**
>
> - Device Posture service supports only PEM issuer certificate type.
> - For the device certificate check on Windows, the EPA client on the end device must be installed with administrative rights. For other checks, you do not require the local administrative rights. For details on the supported scans, see Scans supported by device posture.
> - To install the EPA client with administrative rights on Windows, run the following command in the location where the EPA client plug-in is downloaded.
>
>   ```
>   msiexec /i epasetup.msi
>   ```
>
> - The device certificate check with the Device Posture service does not support the certificate revocation check.
> - If a device certificate is signed by an intermediate certificate, then you must upload the complete chain containing the root and the intermediate certificates in a single PEM file.
>
>   ```
>   Example: chain.pem
>
>   -----BEGIN CERTIFICATE-----
>   ******************************
>   -----END CERTIFICATE-----
>   -----BEGIN CERTIFICATE-----
>   ******************************
>   -----END CERTIFICATE
>   ```

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

Upload device certificate

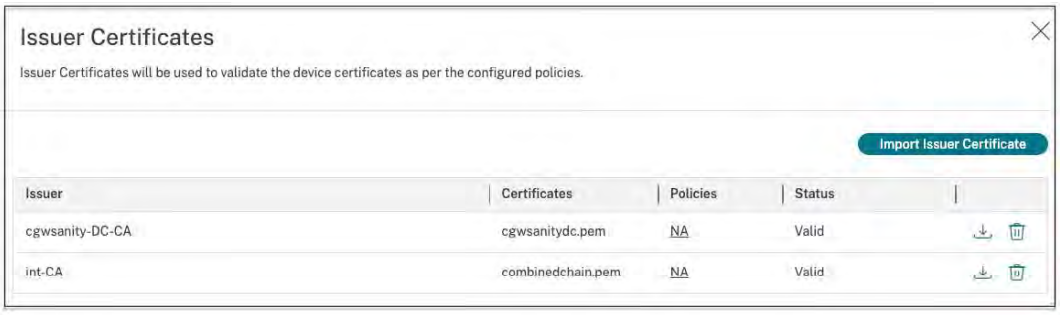
1. Click **Settings** on the Device Posture home page.
2. Click **Manage**, and then click **Import Issue Certificate**.
3. In **Certificate Type**, select the certificate type. Only the PEM type is supported.
4. In **Certificate File**, click **Choose Certificate** to select the issuer certificate.
5. Click **Open**, and then click **Import**.



The selected certificate is listed in **Settings > Issuer Certificates**. You can import multiple certificates.

View imported certificates

1. Click **Settings** on the Device Posture home page.
2. In **Issuer Certificates**, click **Manage**.
3. The Issuer Certificates page lists the imported issuer certificates.



© 1999–2025 Cloud Software Group, Inc. All rights reserved.

Install the device certificate on the end device

Windows:

1. From the **Start** menu, open **Computer Certificate manager**.
2. Ensure that the certificate is installed in `Certificates - Local Computer\Personal\Certificates`.
   - The **Intended Purposes** must include **Client Authentication**.
   - The **Issued By** column must match the issuer name configured on the admin GUI.

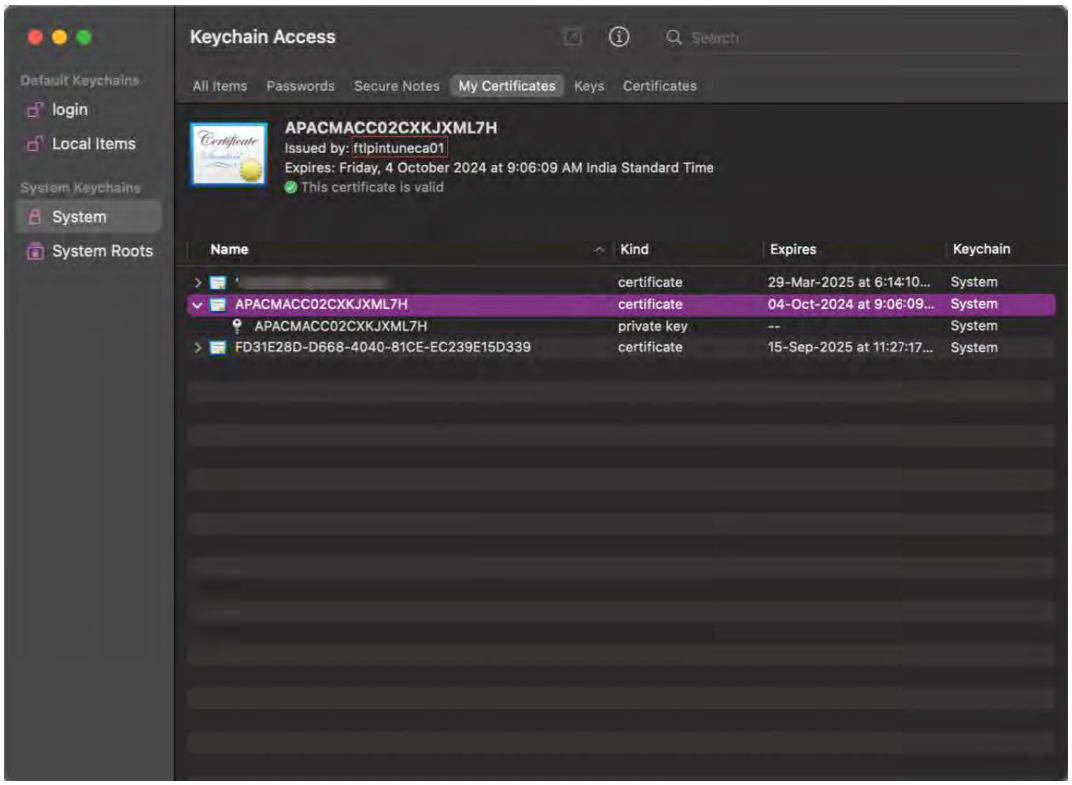


macOS:

1. Open **Keychain Access** and then select **System**.
2. Click **File > Import items** to import the certificate.

   The **Issued by** field must display the certificate issuer name.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.



Third-party integration with device posture

December 3, 2024

In addition to the native scans offered by the Device Posture service, the Device Posture service can also be integrated with the following third-party solutions on Windows and macOS.

- Jamf Pro
- Microsoft Intune
- CrowdStrike

Jamf Pro integration with Device Posture

Jamf Pro, an Apple mobile device management (MDM) software and security provider, helps organizations configure and secure their Apple devices.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

> **Important:**
>
> For the Jamf Pro integration with Device Posture to work on iOS devices, administrators must push the Citrix Workspace app from the Jamf Pro portal. For details, see Push the Citrix Workspace app from the Jamf Pro portal.

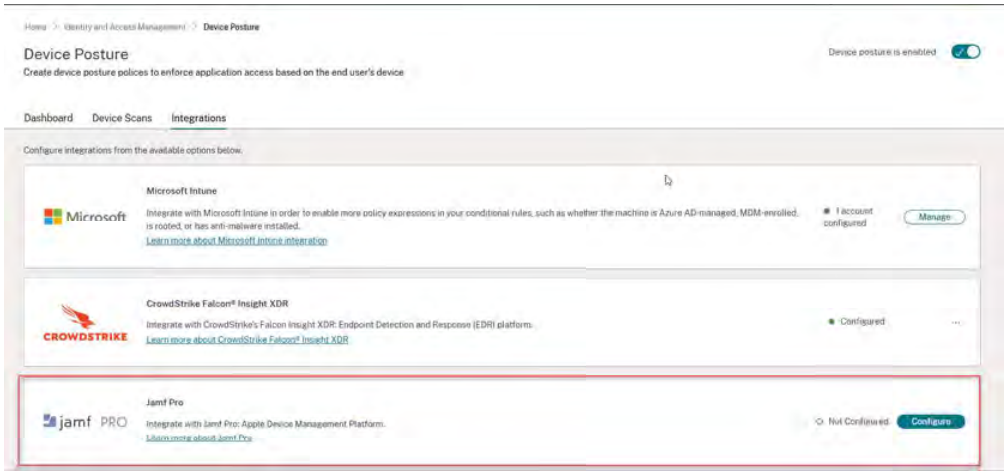
Configure Jamf Pro integration

Jamf Pro integration configuration is a two-step process.

1. Establish trust between Citrix Device Posture service and Jamf Pro service.
2. Configure policies to use Jamf Pro information.

Establish trust between Citrix Device Posture service and Jamf Pro service Perform the following steps to establish trust between Citrix Device Posture service and Jamf Pro service.

1. Sign into Citrix Cloud, and then select **Identity and Access Management** from the menu.
2. Click the **Device Posture** tab, and then click **Manage**.
3. Click the **Integrations** tab.



> **Note:**
>
> Alternatively, customers can navigate to the **Device Posture** option on the left navigation pane of the Secure Private Access service GUI, and then click the **Integrations** tab.

4. Click the ellipsis button in the Jamf Pro box, and then click **Connect**. The **Configure Jamf Pro Integration** pane appears.
5. Enter the client ID, client secret, and Jamf Pro URL and then click **Save**.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

> **Note:**
>
> - You can obtain the API client ID from the Jamf Pro portal.
> - Ensure that you select the **Read Computers** and **Read Mobile Devices** scopes with read permissions for establishing the trust.
> - The Jamf Pro URL is provided by Jamf for each customer account. The Jamf Pro URL is in the format `https://<organization name>.jamfcloud.com`.

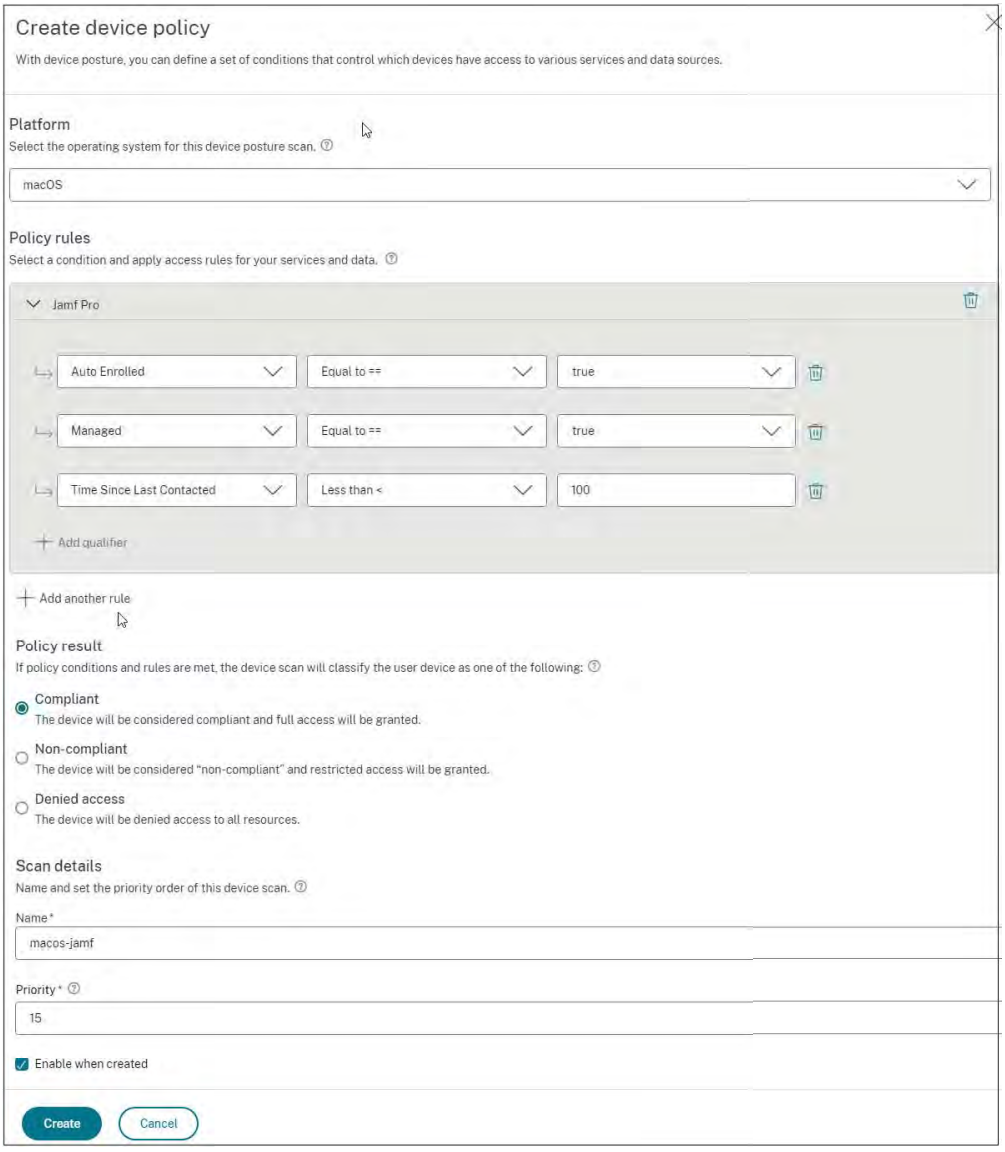
The integration is considered successful after the status changes from **Not Configured** to **Configured**.

If the integration is not successful, the status appears as **Pending**. You must click the ellipsis button, and then click **Reconnect**.

Configure device posture policies

1. Click the **Device Scans** tab and select the platform (**macOS/iOS**) for which this policy is created.
2. Click **Create device policy**.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.



3. In **Policy rules**, select **Jamf Pro**.
4. Select a condition, and then select the values to be matched.

   - **Auto Enrolled -** To check if the Apple device is automatically enrolled into the Jamf system during the initial setup.
   - **Managed -** To check if the device is managed by Jamf Pro.
   - **Time Since Last Contacted -** Applicable for macOS only. To check the time (in minutes) since the device last communicated with the Jamf Pro server.
   - **Time Since Last Inventory Updated -** Applicable for iOS only. To check the time (in min-

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

utes) since the device last communicated with Jamf Pro to update its inventory data.

5. Click **+** to add additional qualifiers.

> **Note:**
>
> You can use this rule with other rules that you configure for Device Posture.

6. In **Policy result** based on the conditions that you have configured, select one of the following.
   - **Compliant**
   - **Non-compliant**
   - **Denied login**
7. Enter the name for the policy and set the priority.
8. Click **Create**.

Push Citrix Workspace app from Jamf Pro portal to iOS devices

For the Jamf Pro integration to work with Device Posture on iOS devices, admins must push the Citrix Workspace app from the Jamf Pro portal to the iOS devices. Perform the following steps:

1. Sign in to your Jamf Pro MDM.
2. Add the Citrix Workspace app that you want to manage.
3. Link the app in the App store.
4. Create an app configuration policy for the app.
5. Add the following XML to the app configuration.

```
<dict>

<key>UDID</key>

<string>$UDID</string>

</dict>
```

Microsoft Intune integration with Device Posture

Microsoft Intune classifies a user's device as compliant or registered based on its policy configuration. During user login into Citrix Workspace, device posture can check with Microsoft Intune about the user's device status and use this information to classify the devices within Citrix Cloud as compliant, non-compliant (partial access), or even deny access to the user login page. Services like Citrix DaaS

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

and Citrix Secure Private Access in turn use device posture's classification of devices to provide contextual access (Smart Access) to virtual apps and desktops, and SaaS and Web apps respectively.

> **Important:**
>
> The Device Posture administrator must use an Intune account with the "Global Administrator" role to configure the Intune integration.

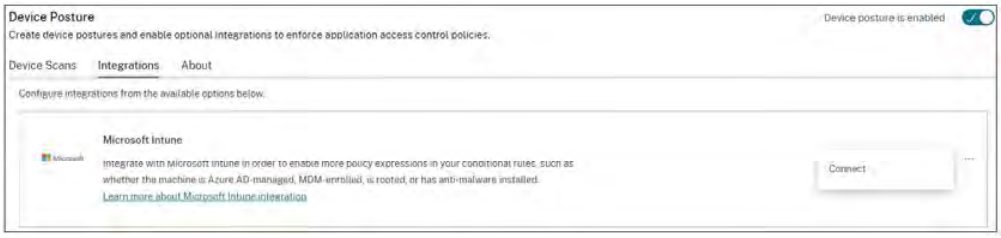
Configure Microsoft Intune integration

Intune integration configuration is a two-step process.

1. Integrate device posture with Microsoft Intune service. This is a one-time activity that you do to establish trust between Device Posture and Microsoft Intune.
2. Configure policies to use Microsoft Intune information.

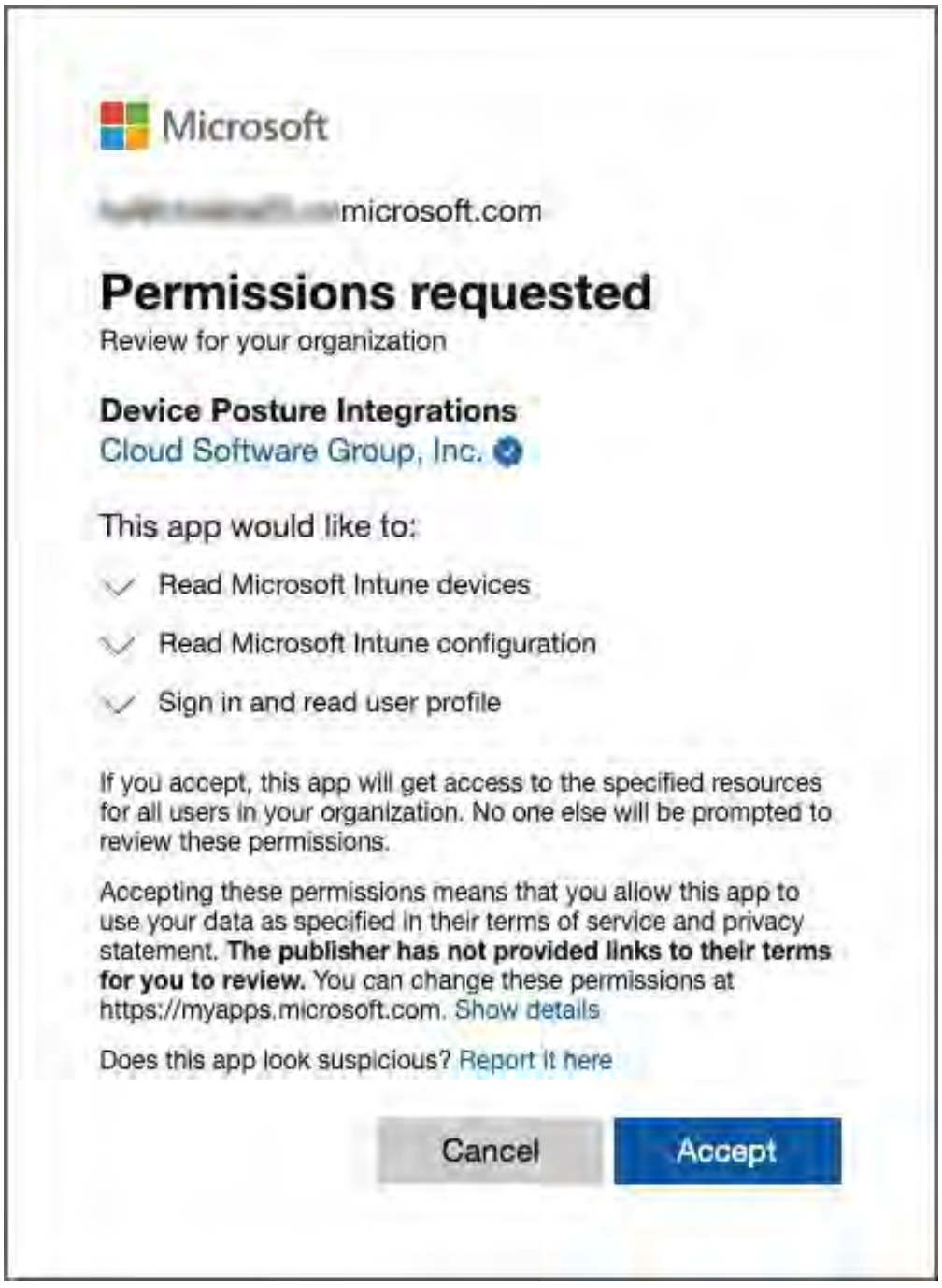
Integrate device posture with Microsoft Intune

1. To access the **Integrations** tab, use one of the following methods:
   - Access the URL https://device-posture-config.cloud.com on your browser, and then click the **Integrations** tab.
   - Secure Private Access customers - On the Secure Private Access GUI, on the left side navigation pane, click **Device Posture,** and then click the **Integrations** tab.



2. Click the **ellipsis** button, and then click **Connect**. The admin is redirected to Azure AD to authenticate.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.



The following table lists the Microsoft Intune API permissions for integration with the Device Posture service.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

API name	Claim value	Permission name	Type
Microsoft Graph	DeviceManagementManagedDevices.Read.All	Read Microsoft Intune devices	Application
Microsoft Graph	DeviceManagementServiceConfig.Read.All	Read Microsoft Intune devices	Application

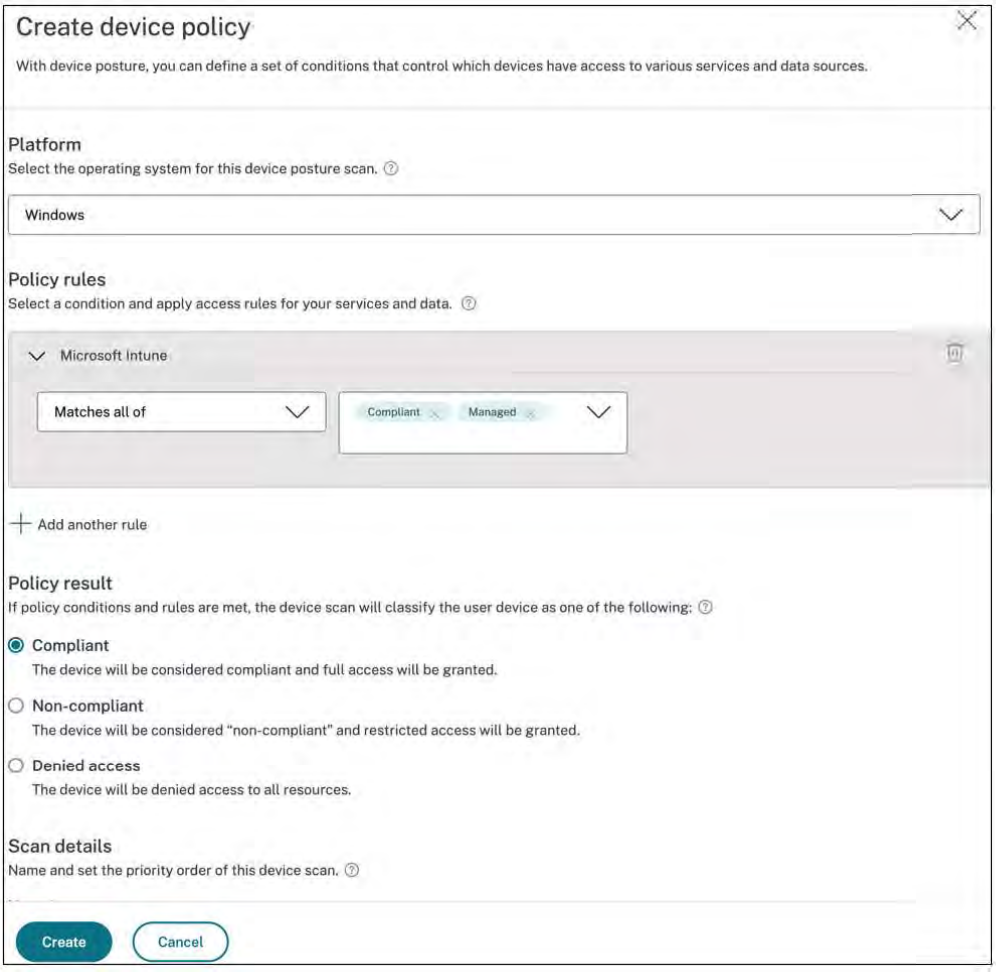
After the integration status changes from **Not Configured** to **Configured**, admins can create a device posture policy.

If the integration is not successful, the status appears as **Pending**. You must click the **ellipsis,** button and then click **Reconnect**.

Configure device posture policies

1. Click the **Device Scans** tab and then click **Create device policy**.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.



2. Enter the name for the policy and set the priority.
3. Select the platform for which this policy is created.
4. In **Policy rules**, select **Microsoft Endpoint Manager**.
5. Select a condition, and then select the MEM tags to be matched.

    - **For Matches any of**, an OR condition is applied.
    - **For Matches all of**, an AND condition is applied.

    > **Note:**
    >
    > You can use this rule with other rules that you configure for device posture.

6. In **Then the device is:** based on the conditions that you have configured, select one of the following.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

- **Compliant (full access is granted)**
- **Non-compliant (Restricted access is granted)**
- **Denied login**

For more details about creating a policy, see Configure device posture policy.

CrowdStrike integration with Device Posture

CrowdStrike Zero Trust Assessment (ZTA) delivers security posture assessments by calculating a ZTA security score from 1 to 100 for each end device. A higher ZTA score means that the posture of the end device is better.

Citrix Device Posture Service can enable contextual access (Smart Access) to Citrix Desktop as a Service (DaaS) and Citrix Secure Private Access (SPA) resources by using the ZTA score of an end device.

Device Posture administrators can use ZTA score as part of policies and classify the end devices as compliant, non-compliant (partial access), or even deny access. This classification can in turn be used by organizations to provide contextual access (Smart Access) to virtual apps and desktops, and SaaS and Web Apps. ZTA score policies are supported for Windows and macOS platforms.

Configure CrowdStrike integration

CrowdStrike integration configuration is a two-step process.

1. Establish trust between Citrix Device Posture service and CrowdStrike ZTA service. This is a one-time activity.
2. Configure access policies. The access policies use the CrowdStrike ZTA score as a rule to provide smart access to Citrix DaaS and Citrix Secure Private Access resources.

Establish trust between Citrix Device Posture service and CrowdStrike ZTA service Perform the following steps to establish trust between Citrix Device Posture service and CrowdStrike ZTA service.

1. Sign into Citrix Cloud, and then select **Identity and Access Management** from the hamburger menu.
2. Click the **Device Posture** tab, and then click **Manage**.
3. Click the **Integrations** tab.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.



> **Note:**
>
> Alternatively, customers can navigate to the **Device Posture** option on the left navigation pane of the Secure Private Access service GUI, and then click the **Integrations** tab.

4. Click the ellipsis button in the CrowdStrike box, and then click **Connect**. The CrowdStrike Falcon Insight XDR integration pane appears.
5. Enter the client ID and client secret and then click **Save**.

> **Note:**
>
> - You can obtain the ZTA API client ID and client secret from the CrowdStrike portal (**Support and resources > API clients and keys**).
> - Ensure that you select the **Zero Trust Assessment** and **Host** scopes with read permissions for establishing the trust.

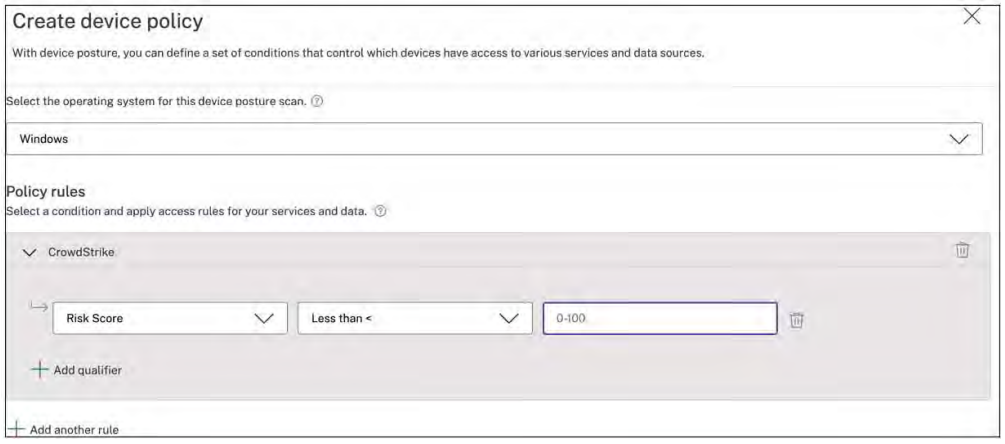
The integration is considered successful after the status changes from **Not Configured** to **Configured**.

If the integration is not successful, the status appears as **Pending**. You must click the ellipsis button, and then click **Reconnect**.

Step 2 - Configure device posture policies Perform the following steps to configure policies to use the CrowdStrike ZTA score as a rule to provide smart access to Citrix DaaS and Citrix Secure Private Access resources.

1. Click the **Device Scans** tab and then click **Create device policy**.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

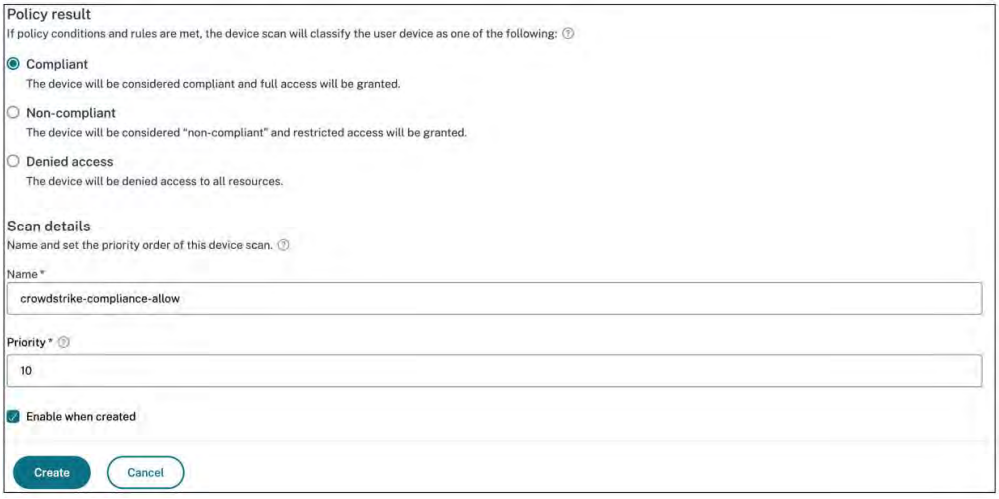


2. Select the platform for which this policy is created.
3. In **Policy Rule**, select **CrowdStrike**.
4. For the **Risk Score** qualifier, select the condition, and then enter the risk score.
5. Click **+** to add a qualifier that checks if the CrowdStrike Falcon sensor is running.

   > **Note:**
   >
   > You can use this rule with other rules that you configure for device posture.

6. In **Policy result** based on the conditions that you have configured, select one of the following.

   - **Compliant**
   - **Non-compliant**
   - **Denied login**



© 1999–2025 Cloud Software Group, Inc. All rights reserved.

7. Enter the name for the policy and set the priority.
8. Click **Create**.

End user flow

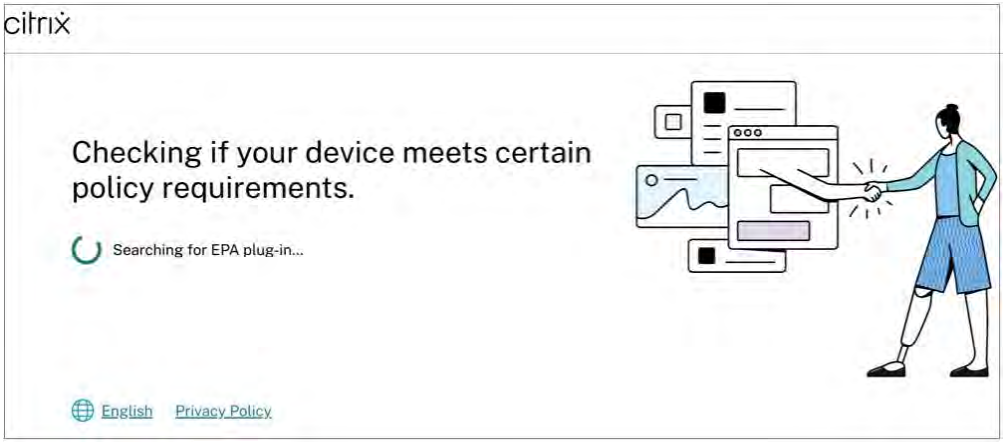
August 27, 2024

Once the device posture policies are set and device posture is enabled, the following are the end-user flows based on how the end user is logging into Citrix Workspace.

End-user flow via browser access

> **Note:**
>
> The macOS client and Chrome browser are used as an example for illustration purposes. The screens and the notifications vary depending on the client and the browser that you use for accessing the Citrix Workspace URL.

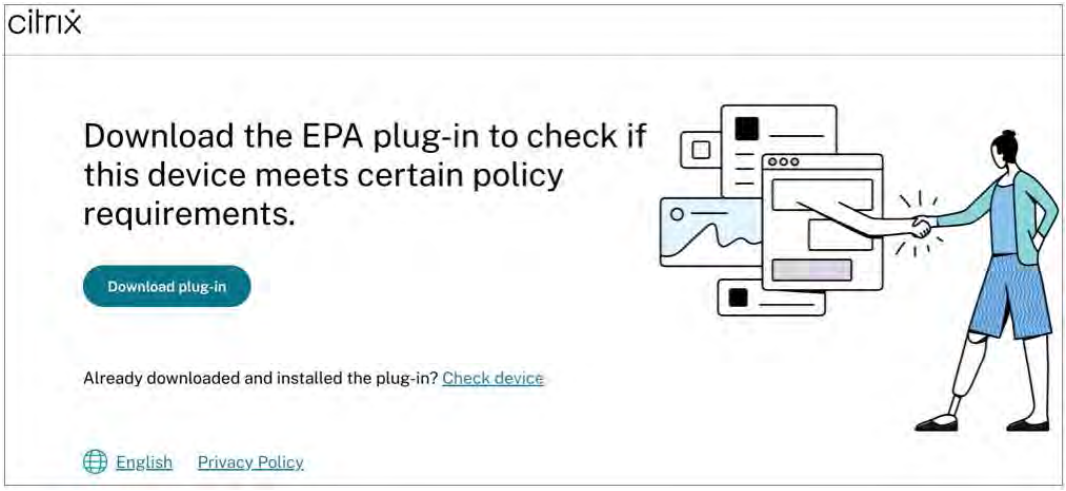
- When an end-user logs on to the Citrix Workspace URL `https://<your-workspace-URL` through a browser, the end user is prompted to run the Citrix EndPointAnalysis application.



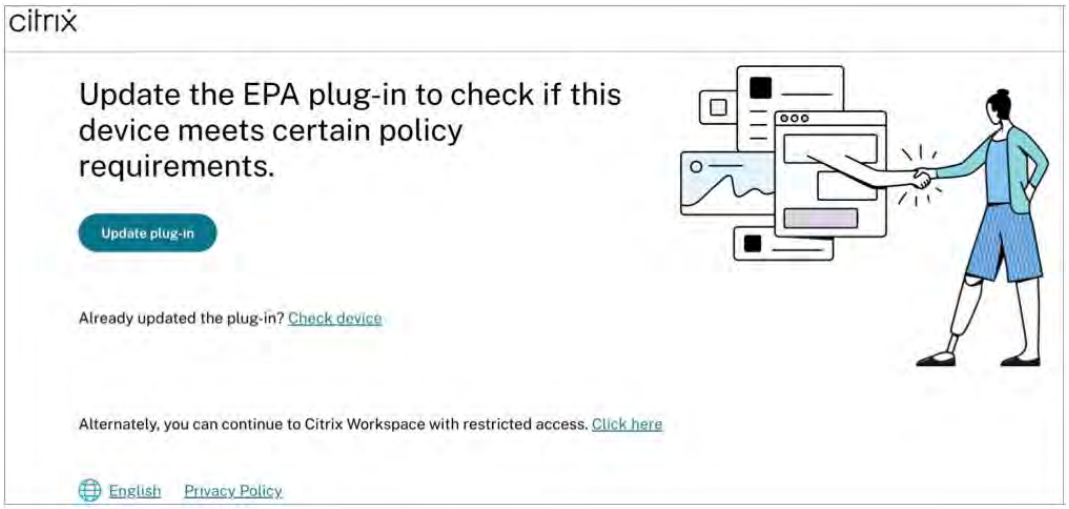
- When the end user clicks **Open Citrix End Point Analysis**, the device posture client runs and scans the endpoint parameters based on device posture policy requirements.
- If the device posture client is not installed on the device, the users are redirected to a page that displays the **Download plug-in** option. If the latest device posture client is already installed on the endpoint, the user must click **Check device** to confirm the same. Similarly, if the EPA installed on the device is not the latest version, the users are redirected to a page that displays

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

the **Update plug-in** option. If the EPA client is already updated, then the user must click **Check device** to confirm the same.



In both scenarios, if the skip check feature is enabled, the message *Alternately, you can continue to Citrix Workspace with restricted access.* is displayed on the **Download plug-in** or the **Update plug-in** pages.



End-user flow via Citrix Workspace application

- When an end-user logs on to the Citrix Workspace URL `https://your-workspace-url` through the Citrix Workspace application, the device posture client installed on the endpoint runs and scans the endpoint parameters based on device posture policy requirements.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

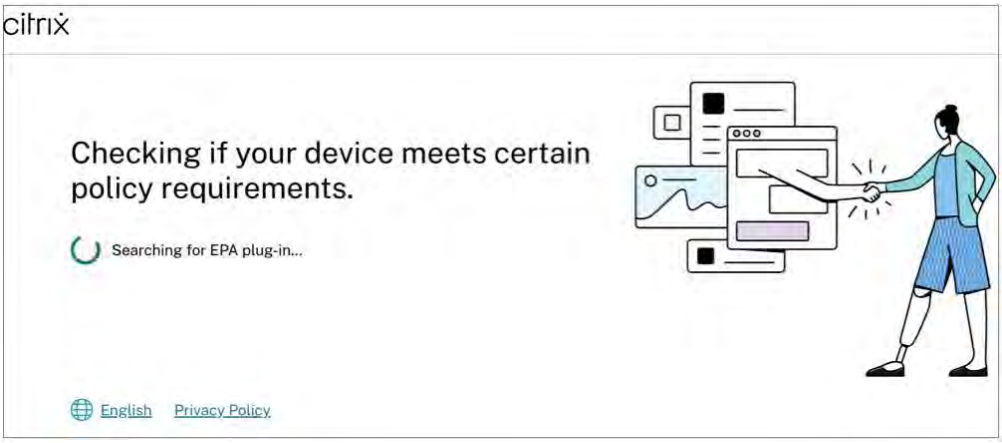
- If the latest device posture client isn't installed on the endpoint, the users are redirected to the page that displays the options **Check again** and **Download Client**. The user must click **Download Client**.
- If the latest device posture client is already installed on the endpoint, the user must click **Check again**.

End-user flow - Device posture results

Based on the device posture policy conditions, three possibilities can occur.

If an endpoint meets the policy conditions such that the device is categorized as;

- **Compliant -** The end user is allowed to log in with unrestricted access to Secure Private Access or Citrix DaaS resources.
- **Non-compliant** - The end user is allowed to log in with restricted access to Secure Private Access or Citrix DaaS resources.



If an endpoint meets the policy conditions such that the device is categorized as **Denied access**, the **Access denied** message appears.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.



Device Posture service in test mode

September 23, 2024

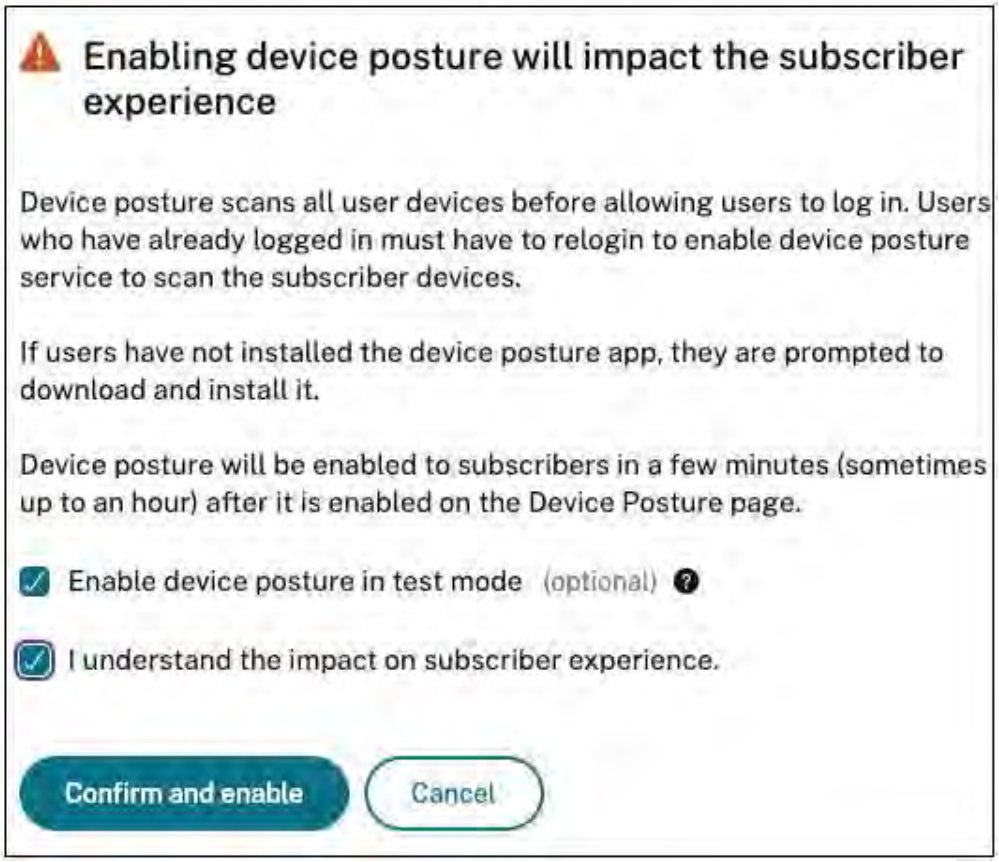
The Device Posture service is also available in test mode wherein admins can test the Device Posture service before enabling it on their production environment. This enables the admins to analyze the impact of the device posture scans on the end user devices and then plan their course of action accordingly before enabling it on production. The Device Posture service in test mode collects data of the end user devices and classifies the devices into the three categories namely, compliant, non-compliant, and denied. However this classification does not enforce any actions on the end user devices. Instead, it empowers administrators to evaluate their environments and enhance security. Admins can view this data on the Device Posture dashboard. Admins can also disable the test mode, if necessary.

> **Note:**
>
> The EPA client must be installed on the devices. In case an end device does not have the EPA client installed, the Device Posture service presents a download page to the end user to download and install the client, without which the end user cannot log on.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

Enable test mode

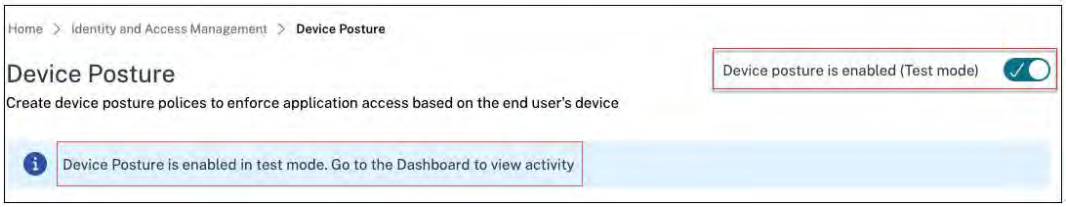
1. Sign in to Citrix Cloud, and then select **Identity and Access Management** from the hamburger menu.
2. Click the **Device Posture** tab and then click **Manage**.
3. Slide the **Device posture is disabled** toggle switch ON.
4. In the confirmation window, select both the checkboxes.



5. Click **confirm and enable**.

When the Device Posture service is enabled in test mode, the Device Posture home page displays a note confirming the same.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.



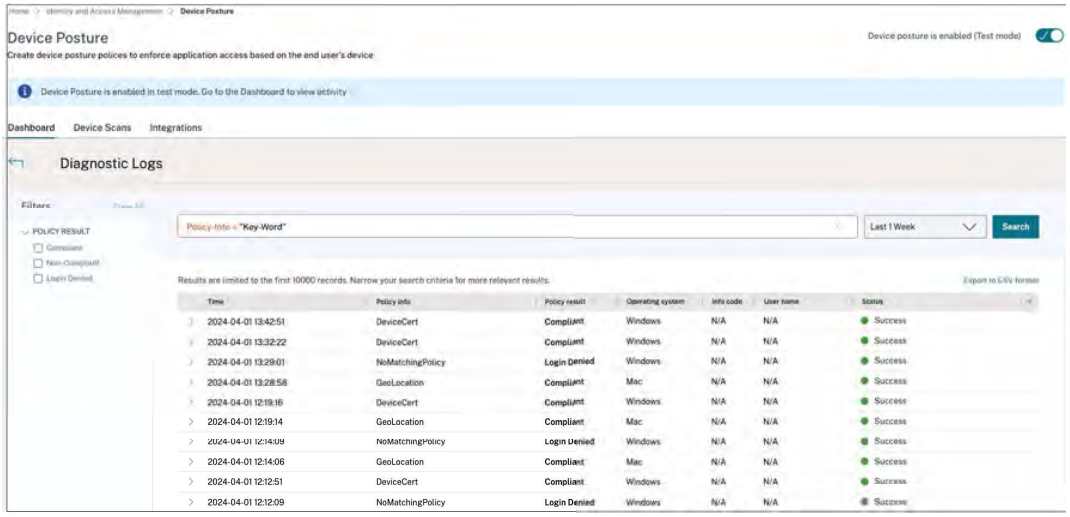
Admins can configure the policies and rules for device posture scans. For details, see Configure device posture. Based on the scan results, the end user devices are classified as compliant, non-compliant, and denied. Admins can view this data on the dashboard.

View the test mode activities on the dashboard

1. Click the **Dashboard** tab on the Device Posture page.

   The **Diagnostic logs** chart displays the number of devices classified as compliant, non-compliant, and login denied.

2. To view the details, click the **See more** link.



Admins can download the monitoring logs from the UI.

Enable test mode in production

If the Device Posture service is already enabled on production, perform the following steps to enable the test mode:

1. On the home page, slide the **Device Posture is enabled** toggle switch OFF.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

2. Select **I understand all device posture checks will be disabled**.
3. Click **confirm and disable**.
4. Now enable the device posture by sliding the **Device Posture is disabled** toggle switch ON.
5. In the confirmation window, select both of the following options.
    - **Enable device posture in test mode**
    - **I understand the impact on subscriber experience**
6. Click **confirm and enable**.

Transition from test mode to production

To transition from test mode to production, you must first disable Device Posture in the test mode and then enable Device Posture again without selecting the option **Enable device posture in test mode**.

> **Important:**
>
> - It is important to thoroughly review your policies before transitioning from test mode to production. Policies that were set up in test mode might behave differently when enforced in production, potentially impacting user access specifically **Deny Access**. In test mode, **Deny Access** is effectively treated as **Non-Compliant,** allowing users to continue accessing the system without disruption. However, in production, this outcome directly blocks access, potentially impacting user experience and operations.
> - Also, when transitioning from test mode to production, there might be potential downtime. It is recommended to plan your transition carefully to minimize disruptions.

Export of Device Posture user events into SIEM systems

December 17, 2024

The Device Posture service events are exported to Security Information and Event Management (SIEM). This allows you to view the following user events associated with the Device Posture service in your SIEM:

- All event types related to Device Posture service.
- Device Posture evaluation events, which provide information about the evaluation of a device's posture against the defined policies.

The following are the benefits of exporting events into SIEM systems:

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

K.Mizra LLC, Ex. 2001, p. 164
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

- Enables your Security Operations teams to correlate, analyze, and search data from disparate logs.
- Helps your Security Operations teams to identify and quickly remediate the security risks.
- Visibility of security alerts in a centralized place.
- Centralized approach to detect potential security threats for organizational risk analysis capabilities such as risk indicators, user profiles, and risk scores.
- Ability to combine and correlate the Citrix Analytics risk intelligence information of a user account with the external data sources connected within your SIEM.

For more information, see the following topics:

- Device Posture service events
- Diagnose Device Posture service events failure
- Security Information and Event Management (SIEM) integration

Troubleshoot Device Posture issues using DaaS Monitor

December 17, 2024

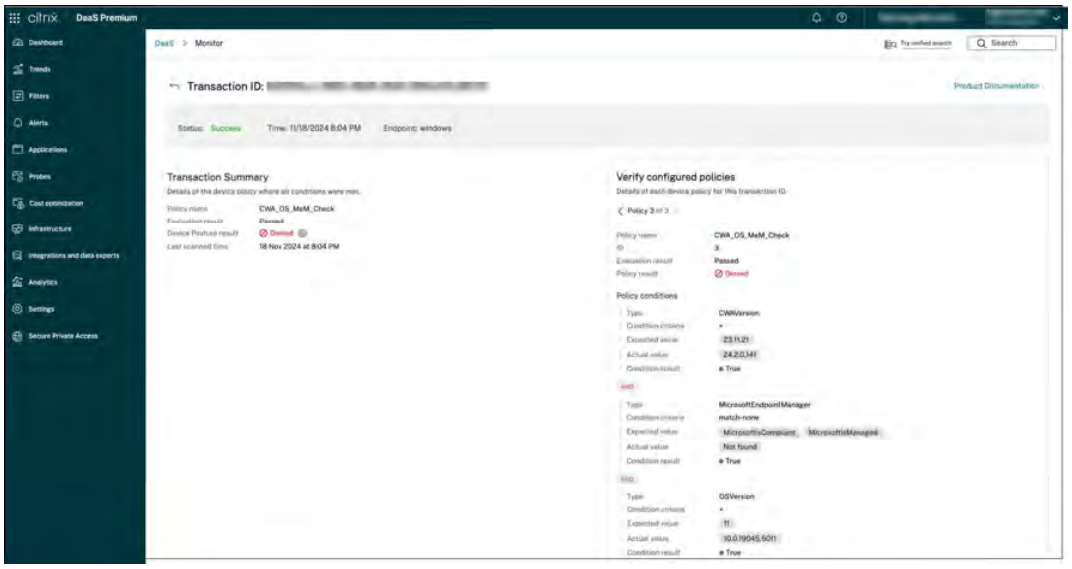
You can troubleshoot the Device Posture issues/errors in DaaS Monitor using the transaction ID that is generated for each session when a user connects through Secure Private Access or DaaS. When you locate the transaction ID, the DaaS Monitor displays information about the session and the associated Device Posture policy that was applied. Monitor also displays the policy results, including whether the device passed or failed the posture checks.

Device posture events on Citrix DaaS Monitor

Perform the following steps to view the events logs for the Device Posture service.

1. Copy the transaction ID of the failed or access denied session from the end user device.
2. Sign into Citrix Cloud.
3. On the DaaS tile, click **Manage**, and then click the **Monitor** tab.
   In the Monitor UI, search for the 32-digit transaction ID and click **Details**.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.



For more information, see the following topics:

- Device Posture service events
- Diagnose Device Posture service events failure

Device Posture logs and events

December 17, 2024

You can view the Device Posture logs and events in the Secure Private Access dashboard and in the Device Posture dashboard in the Identity and Access Management console.

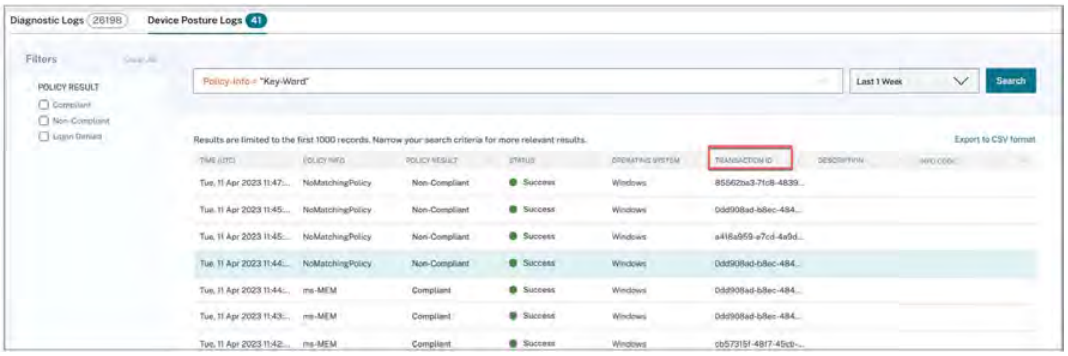
Secure Private Access dashboard

Perform the following steps to view the logs and events for the Device Posture service.

1. Sign into Citrix Cloud.
2. On the Secure Private Access tile, click **Manage** and then click **Dashboard**.

   The **Logging and Troubleshooting** section displays the diagnostic logs related to the Device Posture service.
3. Click the **See more** link to view the details of the logs. You can refine your search based on the policy results (**Compliant**, **Non-Compliant**, and **Login Denied**).

© 1999–2025 Cloud Software Group, Inc. All rights reserved.



- Admins can filter the logs and events based on the transaction ID in the Diagnostic logs chart. The transaction ID is also displayed to the end user whenever access is denied.
- If there's an error or a scan failure, the Device Posture service displays a transaction ID. This transaction ID is available in the Secure Private Access service dashboard. If the logs do not help resolve the issue, end users can share the transaction ID with Citrix Support to resolve the issue.
- The Windows client logs can be found at:
  - %localappdata%\Citrix\EPA\dpaCitrix.txt
  - %localappdata%\Citrix\EPA\epalib.txt
- The macOS client logs can be found at:
  - ~/Library/Application Support/Citrix/EPAPlugin/EpaCloud.log
  - ~/Library/Application Support/Citrix/EPAPlugin/epaplugin.log

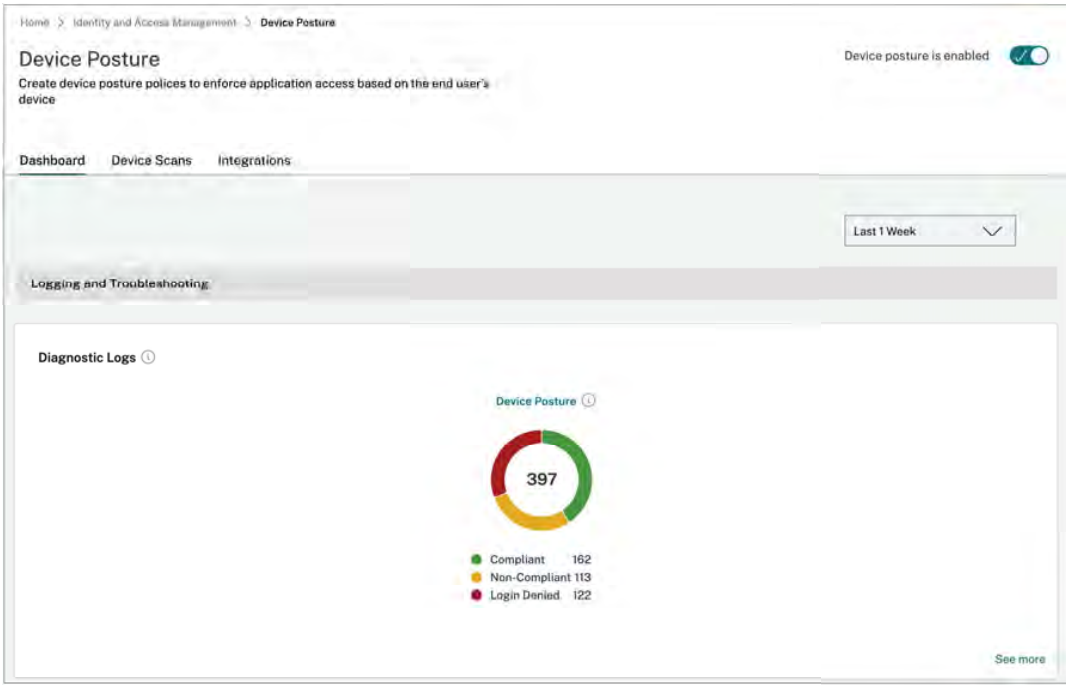
Device Posture dashboard in Identity and Access Management console

Perform the following steps to view the logs and events for the Device Posture dashboard in Identity and Access Management console.

1. Sign into Citrix Cloud.
2. From the Citrix Cloud menu, select **Identity and Access Management**,
3. Click **Device Posture > Manage** and then click **Dashboard.**

   The **Logging and Troubleshooting** section displays the diagnostic logs related to the Device Posture service.
4. Click the **See more** link to view the details of the logs. You can refine your search based on the policy results (**Compliant**, **Non-Compliant**, and **Login Denied**).

© 1999–2025 Cloud Software Group, Inc. All rights reserved.



Device posture error logs

The following logs related to the Device Posture service can be viewed on the Citrix Monitor and Secure Private Access dashboard. For all these logs, it's recommended that you contact Citrix Support for resolution.

- Failed to read configured policies
- Failed to evaluate endpoint scans
- Failed to process policies/expression
- Failed to save endpoint details
- Failed to process scan results from endpoints

Manage Citrix Endpoint Analysis client for Device Posture service

December 16, 2024

Citrix Device Posture service is a cloud-based solution that helps admins to enforce certain requirements that the end devices must meet to gain access to Citrix DaaS (virtual apps and desktops) or Citrix Secure Private Access resources (SaaS, Web apps, TCP, and UDP apps).

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

To run device posture scans on an end device, you must install the Citrix EndPoint Analysis (EPA) client, which is a lightweight application, on that device. Device Posture service always runs with the latest version of the EPA client released by Citrix.

Installation of the EPA client

During runtime, the Device Posture service prompts the end user to download and install the EPA client during run-time. For details, see End-user flow.
Usually, an EPA client does not require local admin rights to download and install on an endpoint. However, to run device certificate check scans on an end device, the EPA client must be installed with administrator access. For details about installing an EPA client with administrator access, see Install device certificate on the end device.

Upgrade of the EPA client for Windows

When a new version of the EPA client is released, the EPA clients for Windows are upgraded by default after the first installation. Auto-upgrade ensures that the end-user devices are always running on the latest version of the EPA client that is compatible with the Device Posture service. For the auto-upgrade, the EPA client must have been installed with administrator access.

Distribution of the EPA client

EPA clients can be distributed using Global App Configuration service (GACS) or EPA integrated with Citrix Workspace app installer, or using software deployment tools.

- **EPA client installer integrated with Citrix Workspace app:** The EPA client installer is integrated with Citrix Workspace app 2402 LTSR for Windows and macOS. This integration eliminates the need for the end users to install EPA client separately after installing Citrix Workspace app.
  - To install the EPA client as part of Citrix Workspace app for Windows, use the command line option `InstallEPAClient`. For example, `./CitrixworkspaceApp.exe InstallEPAClient`.
  - For macOS, the EPA client installer is installed by default.

> **Note:**
>
> - EPA client installation as part of Citrix Workspace app is disabled by default. It must be explicitly enabled by using the command line option `InstallEPAClient`.
> - If an end device already has an EPA client installed and the end user installs Citrix Work-

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

space app, the existing EPA client is upgraded.
- If an end user uninstalls Citrix Workspace app, then the integrated EPA client is also removed from the device, by default. However, if the EPA client was not installed as part of the integrated Citrix Workspace app installation, then the existing EPA client is retained in the device.
- The EPA client installer integrated with Citrix Workspace app can also be used with NetScaler. For details, see Manage EPA client when used with NetScaler and Device Posture.

- **Distribute the client using GACS**: GACS is a Citrix provided solution to manage the distribution of client-side agents (plug-ins). The Auto update service available in GACS ensures that the end devices are on the latest EPA versions without end user intervention. For more information on GACS, see How to use the Global App Configuration service.

Note:

- GACS is supported on Windows devices only for distributing the EPA client.
- To manage an EPA client through GACS, install Citrix Workspace Application (CWA) on the end devices.
- If CWA is installed with administrator privileges on an end user device, then GACS installs the EPA client with the same administrator privileges.
- If CWA is installed with user privileges on an end user device, then GACS installs the EPA client with the same user privileges.

Distribute the client using Software deployment tools: The latest EPA client can be distributed by admins through software deployment tools like Microsoft SCCM.

Manage EPA client when used with NetScaler and Device Posture

The EPA client can be used together with NetScaler and Device Posture in the following deployments:

- NetScaler based adaptive authentication with EPA
- NetScaler based on-premises gateway with EPA

The Device Posture service pushes the latest version of the EPA client to the end devices. However, on NetScaler, administrators can configure the following version control for the EPA scans on gateway virtual servers:

- **Always**: The EPA client on the end device and NetScaler must be on the same version.
- **Essential**: The EPA client version on the end device must be within the range configured on NetScaler.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

- **Never**: The end device can have any version of the EPA client.

For more information, see Plug-in behaviors.

Considerations when EPA client is used with NetScaler and Device Posture

When an EPA client is used together with Device Posture Service and NetScaler, there might be scenarios where the end device is running the latest EPA client version whereas NetScaler is on a different version of the EPA client. This might result in a mismatch of the EPA client version on NetScaler and the end device. As a result, NetScaler might prompt the end user to install the EPA client version which is present on NetScaler. To avoid this conflict, we recommend the following configuration changes:

- If you have configured EPA with Adaptive Authentication or on-premises authentication or gateway virtual server, it is recommended that you disable version control of the EPA client on NetScaler. This is done to ensure that the GACS or Device Posture service does not push the latest version of the EPA client to the end devices.
- The EPA version control can be set to **Never** by using the CLI or the GUI. These configuration changes are supported on NetScaler 13.x and later versions.
  - CLI: Use the CLI commands for Adaptive Authentication and on-premises authentication virtual server.
  - GUI: Use the GUI for the on-premises gateway virtual server. For details, see Control upgrade of Citrix Secure Access clients.

Sample CLI commands:

```
add rewrite action <rewrite_action_name> insert_http_header Plugin-Upgrade "\"epa_win:Never;epa_mac:Always;epa_linux:Always;vpn_win:Never;vpn_mac:Always;vpn_linux:Always;\""

add rewrite policy <rewrite_action_policy> "HTTP.REQ.URL.CONTAINS(\"pluginlist.xml\")" <rewrite_action_name>

bind authentication vserver <Authentication_Vserver_Name> -policy <rewrite_action_policy> -priority 10 -type RESPONSE
```

Data Governance

December 22, 2023

This topic provides information regarding the collection, storage, and retention of logs by the Device Posture service. Any capitalized terms not defined in the Definitions sections carry the meaning specified in the Citrix End User Services Agreement.

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

Data residency

The Citrix Device Posture customer content data resides in the AWS and Azure Cloud Services. They are replicated to the following regions for availability and redundancy:

- AWS
  - East US
  - West India
  - Europe (Frankfurt)
- Azure
  - West US
  - West Europe
  - Asia (Singapore)
  - South Central US

The following are the different destinations for the service configuration, runtime logs and events.

- Splunk service for system monitoring and debug logs, in the US location only.
- Citrix Analytics Service for the diagnostics and user access logs, see Citrix Analytics Service Data Governance for more information.
- Citrix Cloud System Logs service for admin audit logs. For details, see Citrix Cloud Services Customer Content and Log Handling and Geographical Considerations.

Data collection

Citrix Device Posture service allows the customer administrators to configure the service through the Device Posture UI. The following customer content is collected based on the device posture policy configuration and the platform:

- Operating system version
- Citrix Workspace app version
- MAC addresses
- Running processes
- Device certificate
- Registry details
- Windows installation update details
- Last Windows update details
- File system –file names, file hashes and modified time
- Domain name

For runtime logs collected by the service components, the key information consists of the following:

© 1999–2025 Cloud Software Group, Inc. All rights reserved.

- Customer/tenant ID
- Device ID (Citrix generated unique identifier)
- Device posture scan output
- Public IP address of the endpoint device

Data transmission

Citrix Device Posture service sends logs to destinations protected by transport layer security.

Data control

Citrix Device Posture service does not currently provide options for the customers to turn off sending logs or prevent customer content from being replicated globally.

Data retention

Based on the Citrix Cloud data retention policy, the customer configuration data are purged from the service 90 days after subscription has expired.

The log destinations maintain their service-specific data retention policy.

- For details, see Data Governance for the retention policy for the Analytics logs.
- The Splunk logs are archived and eventually removed after 90 days.

Data export

There are different data export options for different types of logs.

- The admin audit logs are accessible from the Citrix Cloud System Log console.
- The Device posture service diagnostics logs can be exported from the Citrix Analytics Service or Secure Private Access service dashboard as a CSV file.

Definitions

- Customer Content means any data uploaded to a customer account for storage or data in a customer environment to which Citrix is provided access to perform Services.
- Log means a record of events related to the services, including records that measure performance, stability, usage, security, and support.
- Services mean that the Citrix Cloud services outlined earlier for the purposes of Citrix Analytics.

K.Mizra LLC, Ex. 2001, p. 173
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

© 2025 Cloud Software Group, Inc. All rights reserved. This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc. This and other products of Cloud Software Group may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at https://www.cloud.com/legal. Citrix, the Citrix logo, NetScaler, and the NetScaler logo and other marks appearing herein are either registered trademarks or trademarks of Cloud Software Group, Inc. and/or its subsidiaries in the United States and/or other countries. Other marks are the property of their respective owner(s) and are mentioned for identification purposes only. Please refer to Cloud SG's Trademark Guidelines and Third Party Trademark Notices (https://www.cloud.com/legal) for more information.

EXHIBIT 10

Device Posture check for Citrix Workspace

Vijaya Raghavan
Principal Product Manager
Citrix Workspace Security

Date

citrix



Case 0:25-cv-60803-XXXX Document 1-10 Entered on FLSD Docket 04/24/2025 Page 2 of 12

K.Mizra LLC, Ex. 2001, p. 176
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

Ex. 10, p. 2

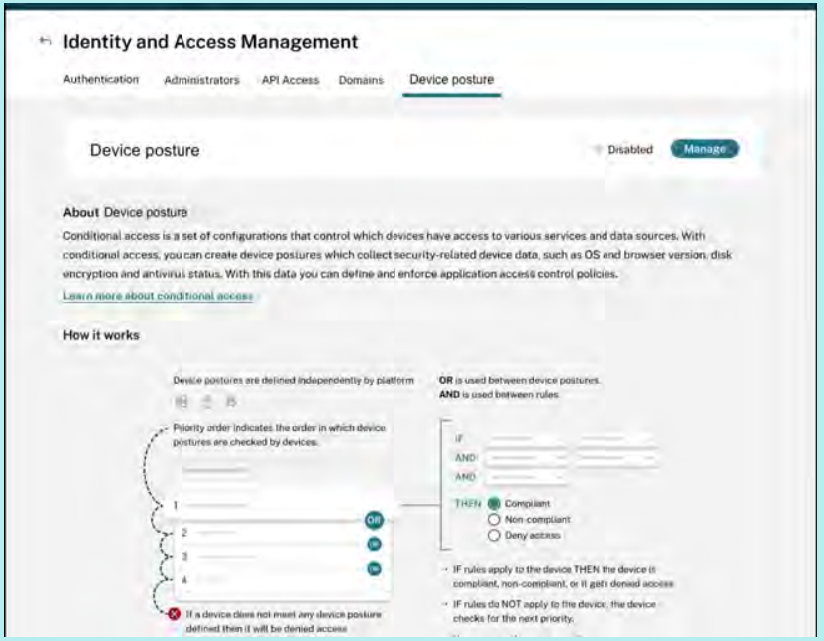
Agenda

- Introduction/Overview
- More about policy builder

© 2021 Citrix | Confidential

Device posture service - Overview

- A solution to establish device trust before allowing user login
- Key pillar to implement a zero-trust based access
- Device posture service will check the device for compliance, if it is managed/BYO and security posture
- An app needs to be installed on client's device to check the posture
- In this phase we will check device compliance by scanning few OS level settings
- This is like EPA of on-prem Gateway, but running natively on cloud and evolve to provide more advanced capabilities



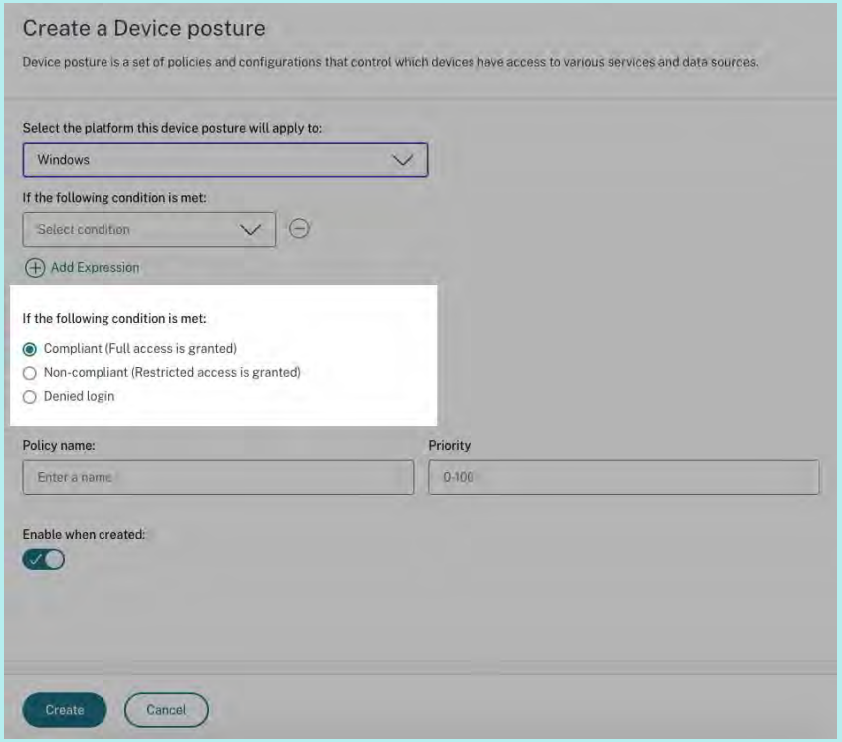
***** Can be used with any authentication method***

© 2021 Citrix | Confidential

Case 0:25-cv-60803-XXXX Document 1-10 Entered on FLSD Docket 04/24/2025 Page 4 of 12
K.Mizra LLC, Ex. 2001, p. 178
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

Device posture service - Capabilities

- Scans supported in first phase: OS checks like version, registry, firewall, file system, MAC address, etc on Win & macOS devices only
- Based on scan result, users/devices can be
  - **Denied login**
  - Allowed login and classified as "**Compliant**" or "**Non-Compliant**"
- The classification of "Compliant" and "Non-compliant" devices is passed onto DaaS and SPA service which can in turn use it to provide contextual access/SmartAccess

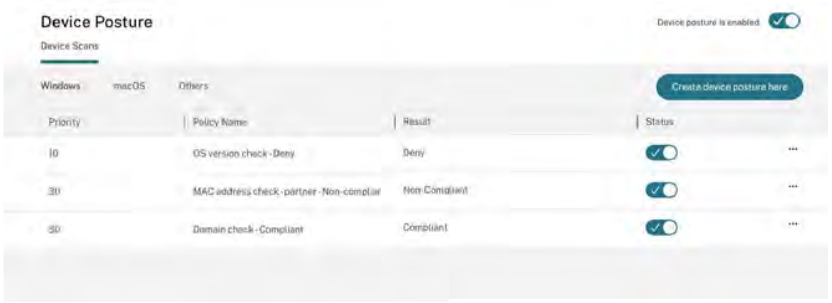


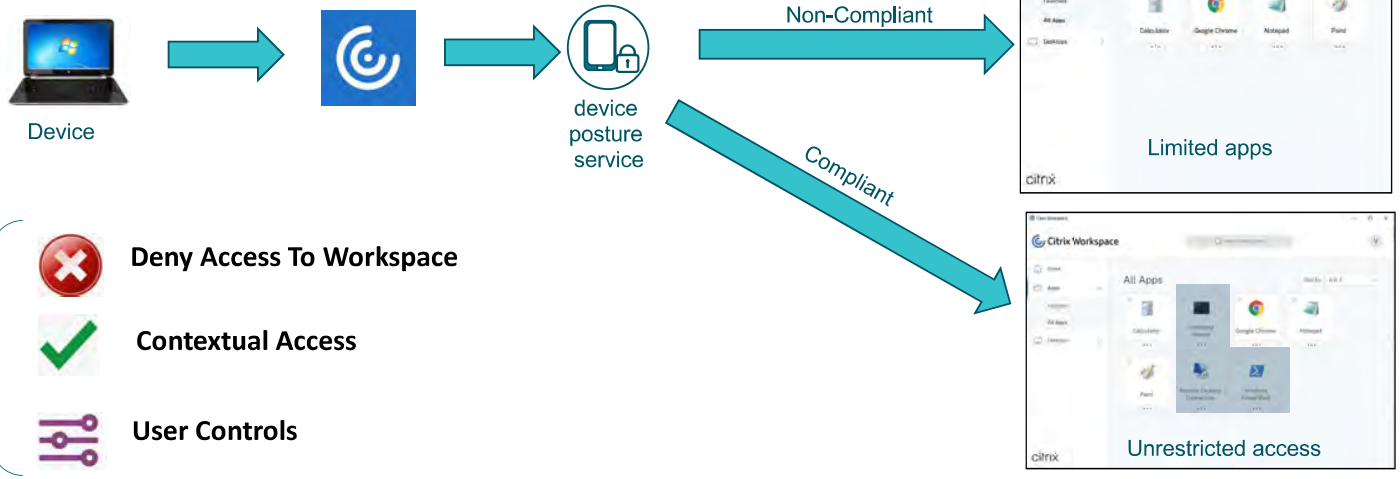
© 2021 Citrix | Confidential

Ex. 10, p. 4

Case 0:25-cv-60803-XXXX Document 1-10 Entered on FLSD Docket 04/24/2025 Page 5 of 12
K.Mizra LLC, Ex. 2001, p. 179
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

Sample use case





© 2021 Citrix | Confidential

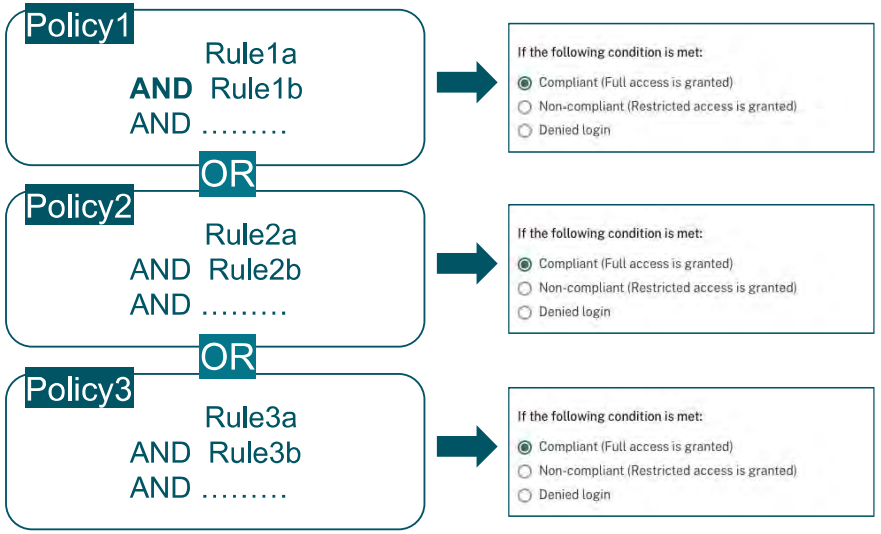
citrix

Case 0:25-cv-60803-XXXX Document 1-10 Entered on FLSD Docket 04/24/2025 Page 6 of 12

K.Mizra LLC, Ex. 2001, p. 180
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

Policy builder

- A combination of Policies and Rules
  - Set of Rules make up a policy
  - Set of Policies make up entire device posture
- Each policy can be attached with below action
  - Denied login
  - Compliant
  - Non-Compliant
- Each policy has priority. Policy evaluation stops if a policy evaluates to TRUE and associated action is taken
- Policies must be set for Windows and macOS separately



© 2021 Citrix | Confidential

citrix

Case 0:25-cv-60803-XXXX Document 1-10 Entered on FLSD Docket 04/24/2025 Page 7 of 12
K.Mizra LLC, Ex. 2001, p. 181
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

Recommendations for policy builder

- Configure Deny policies with higher priority, followed by non-compliant and finally compliant ones.
- Tags/Policies can be used to configure Contextual access on SPA or DaaS



- If none of the policy evaluates to TRUE, by default the device is marked Non-Compliant and allowed to login
- If users try to access from an unsupported platform (other than Windows & macOS) they are marked Non-Compliant and allowed to login
  - Will be customizable to either Deny or Non-compliant by GA

© 2021 Citrix | Confidential

Case 0:25-cv-60803-XXXX Document 1-10 Entered on FLSD Docket 04/24/2025 Page 8 of 12
K.Mizra LLC, Ex. 2001, p. 182
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

Additional info

- You can use Device posture service with Gateway or Adaptive Auth as IDP. However, do not enable EPA on Gateway or Adaptive Auth and use only device posture service
- The Device posture in cloud cannot be bypassed unlike on-prem. We will keep progressively hardening it
- Configuration login page: https://device-posture-config.cloud.com/
- Sign up here for tech preview: https://podio.com/webforms/27886155/2183565.
- Doc: https://citrix.sharefile.com/d-s0ef189bea2e64266b81ac381831943ba

Case 0:25-cv-60803-XXXX Document 1-10 Entered on FLSD Docket 04/24/2025 Page 9 of 12

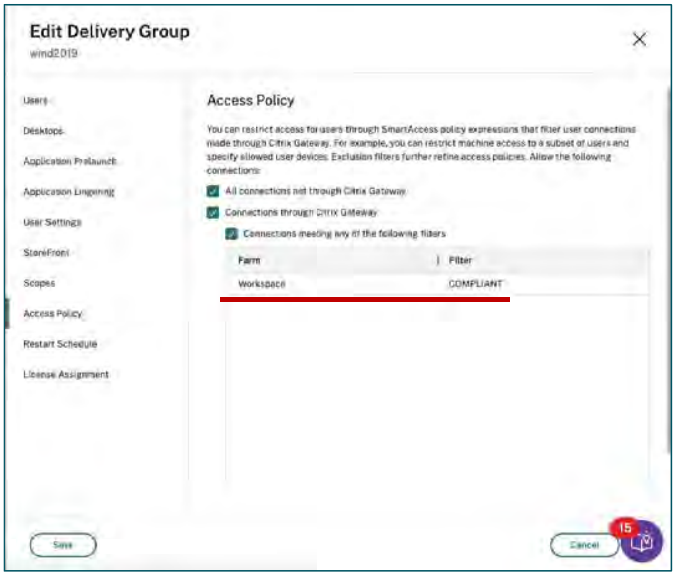
K.Mizra LLC, Ex. 2001, p. 183
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

Contextual access/Smart Access policies

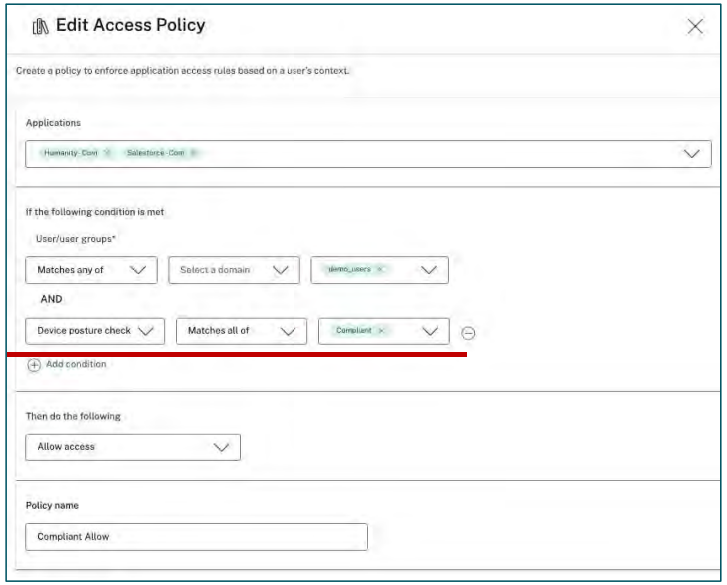
Tags/Policy names to be used for contextual access

SPA	DaaS
Compliant	COMPLIANT
Non-Compliant	NON-COMPLIANT

Sample DaaS config



Sample SPA config



Case 0:25-cv-60803-XXXX Document 1-10 Entered on FLSD Docket 04/24/2025 Page 10 of 12

K.Mizra LLC, Ex. 2001, p. 184
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

Supported Scans in this phase

Windows

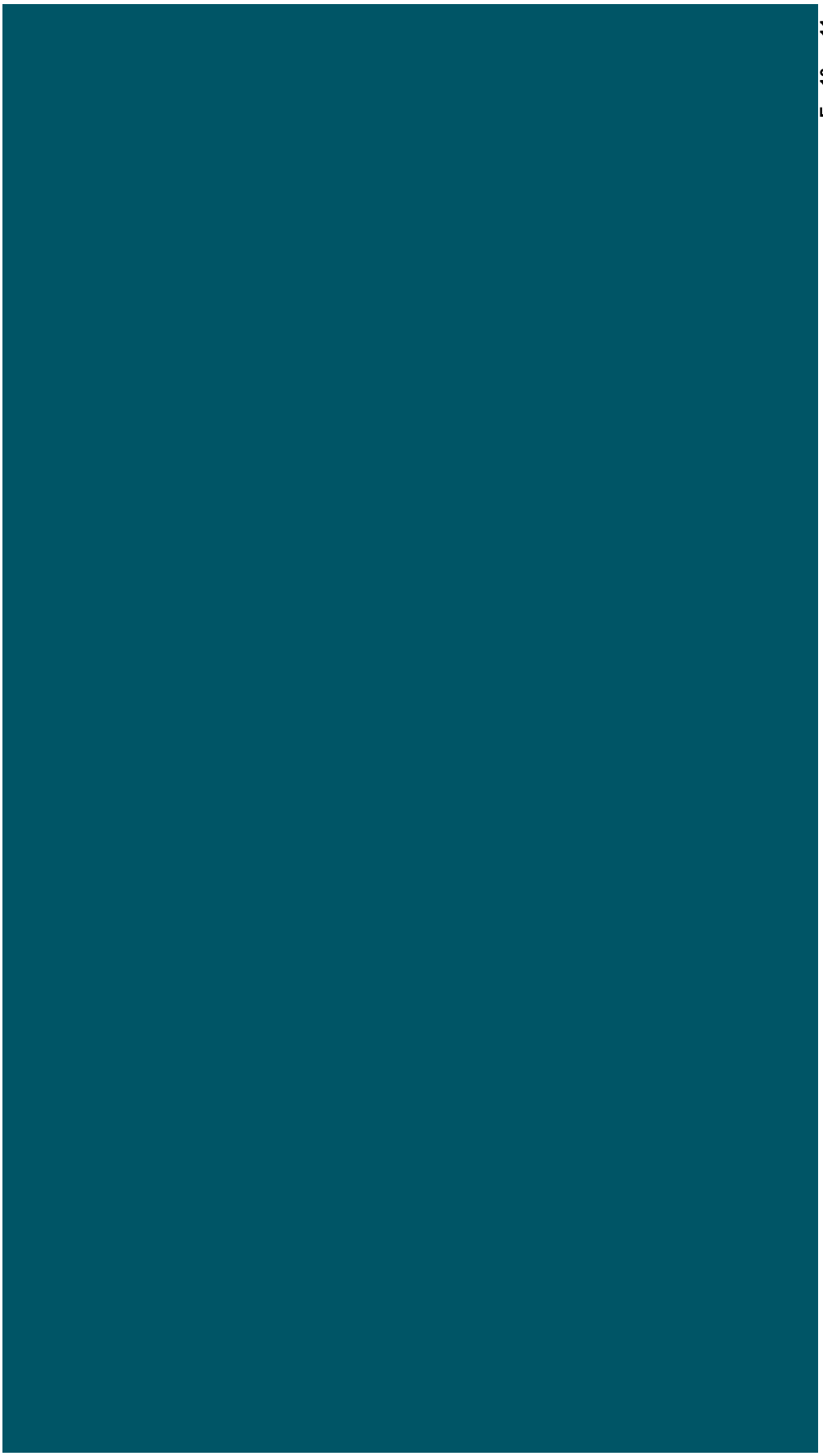
- OS version check
  - Time and date last updated
  - Manual/Automatic update
- File checks including presence of file and few attributes of file like time, hash etc
- Registry key setting
- Domain check
- MAC address check
- Check for processes running

macOS

- OS version check
- File checks including presence of file and few attributes of file like time, hash etc
- MAC address check
- Check for processes running

Case 0:25-cv-60803-XXXX Document 1-10 Entered on FLSD Docket 04/24/2025 Page 11 of 12

K.Mizra LLC, Ex. 2001, p. 185
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468



Ex. 10, p. 11

EXHIBIT 11

February 21, 2025

Contributed by:

IN THIS ARTICLE

Device compliance with event-driven posture checks

Periodic scanning of devices

Customized messages for access denied scenarios

Skip device posture checks

Enable skipping the device posture checks

The following are some of the global Device Posture settings that you can configure as required.

- Device compliance with event-driven posture checks
- Periodic scanning of devices
- Customized messages for access denied scenarios
- Skip device posture checks
- Custom workspace URLs support
- Session Recording configuration with Device Posture
- Device certificate check

Device compliance with event-driven posture checks

Starting with EPA Library version 24.11.1.1, Device Posture introduces enhanced checks to assess the compliance of your devices.

The following two scans are added to provide more control over device compliance:

- **External Device Connected:** This scan detects if an external USB storage device is connected to the endpoint.
- **Network Settings:** This scan determines if the device is connected to a protected Wi-Fi network.

When any of the following events occur, Device Posture re-evaluates the device's compliance status:

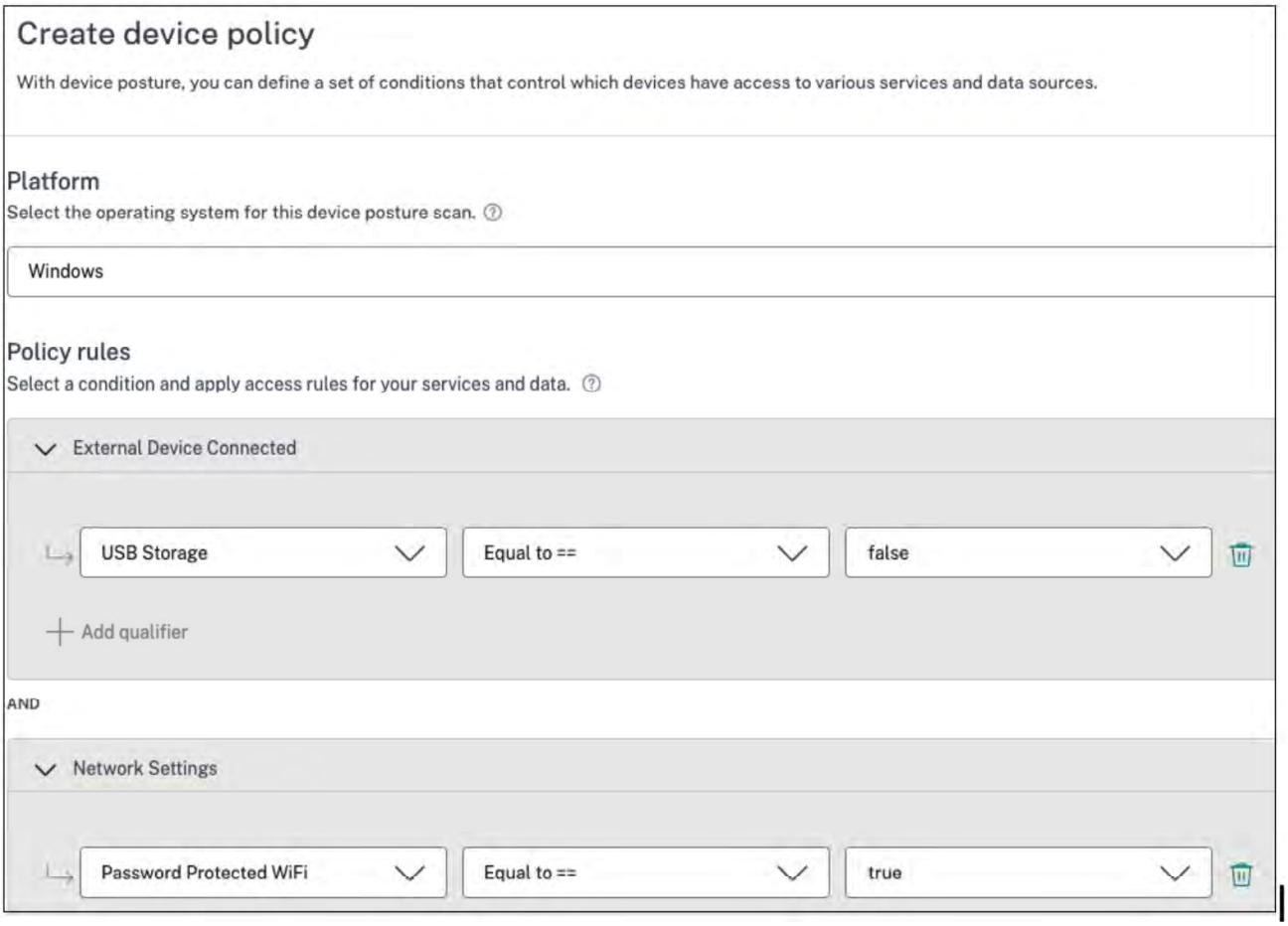
- The **External Device Connected** scan is configured, and an external USB storage device is inserted or removed.
- The **Network Settings** scan is configured, and a switch between an open network and a protected network is detected.
- The **Windows Defender Firewall** scan is configured, and it's enabled or disabled.

Access to the resources is updated in accordance with the new compliance status. This ensures that access to resources is granted or revoked based on the device's real-time compliance status.

NOTE:

Event-driven posture checks are currently supported only on Windows platforms. While both user and administrator installations can utilize the posture scans, real-time re-evaluation of device compliance based on events requires administrator privileges.

The following figure displays the **External Device Connected** and **Network Settings** scans configured in the Device Posture admin console.



Periodic scanning of devices

You can enable periodic scanning of Windows devices for the configured checks every 30 minutes. The EPA client on the end device must be installed with administrative rights to enable periodic scanning of devices. To enable periodic scanning, do the following:

1. Navigate to **Device Posture > Device Scans** and click **Settings**.
2. In **Periodic device posture scans** section, slide the toggle switch ON to enable periodic scanning of devices.



- For HTTP/HTTPS applications, if a device posture scan result changes from **Compliant** to **Non-compliant** or **Denied access**, access to the resources is updated in accordance with the new compliance status.
- For TCP/UDP applications launched from the Citrix Secure Access client, if a device posture scan results in a downgrade (for example from **Compliant** to **Non-compliant** or **Denied access**), users receive a notification, and after 5 minutes, the active sessions are terminated.



Customized messages for access denied scenarios

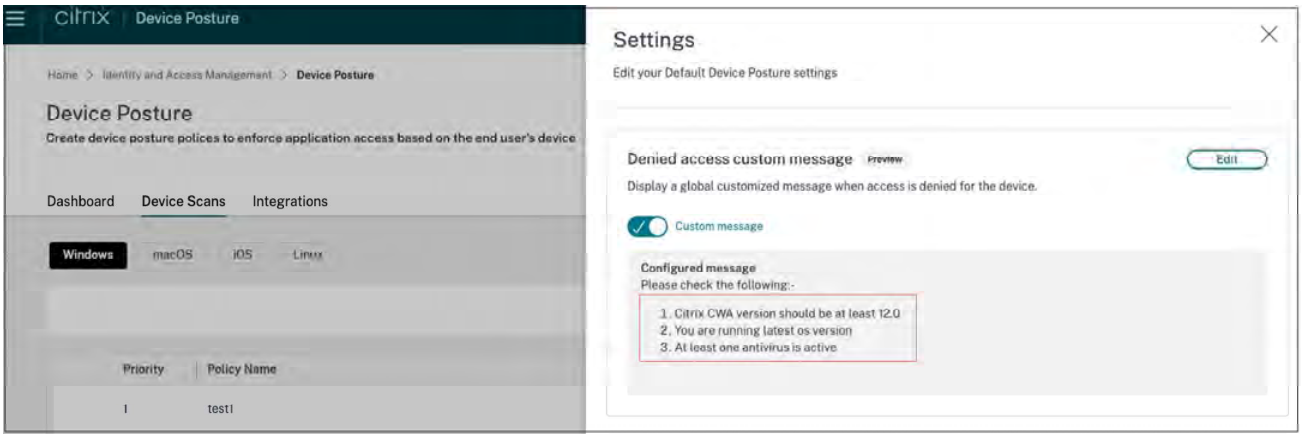
Admins can customize the message that is displayed on the end device when access is denied.

Perform the following steps to add customized messages:

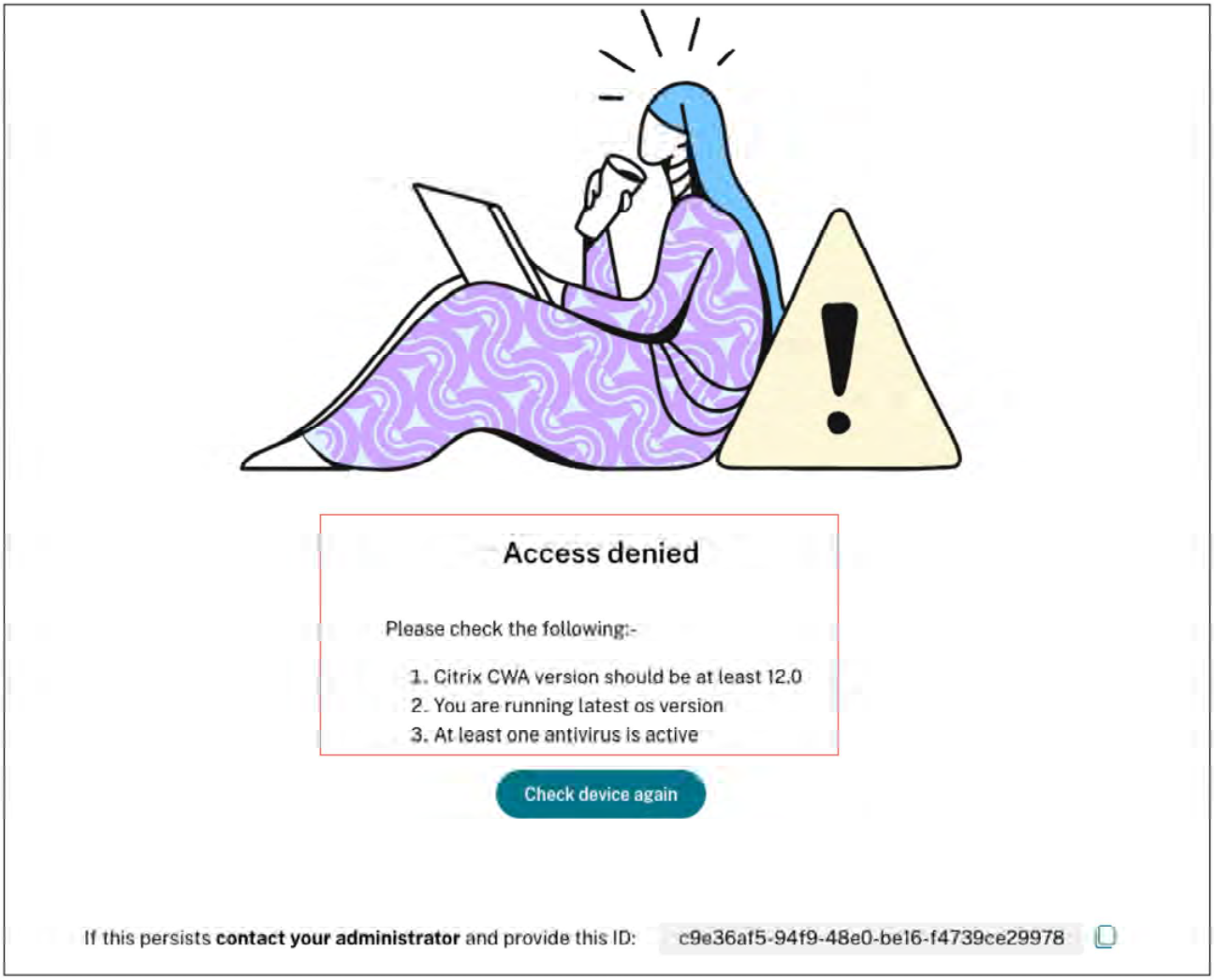
1. Navigate to the **Device Posture > Device Scans** page.
2. Click **Settings**.
3. Click **Edit** and in the **Message** box, enter the message that must be displayed in access denied scenarios. You can enter a maximum of 256 characters.
4. Click **Enable custom message on save** to enforce the option of displaying the custom message. If you do not select this checkbox, the custom message is created but not displayed on the devices in access denied scenarios.

   Alternatively, you can enable the **Custom message** toggle switch on the **Settings** page to display the message on the devices.
5. Click **Save**.

The following image displays a sample message added by the admin.



The following image displays the custom message that appears on the end user device when access is denied.



The message that you have entered appears whenever access is denied for the end device.

Skip device posture checks

Admins can allow the end users to skip the device posture checks on their devices in the following scenarios:

- The device posture agent is not installed on the device.
- The EPA client installed on the device is not the latest version.

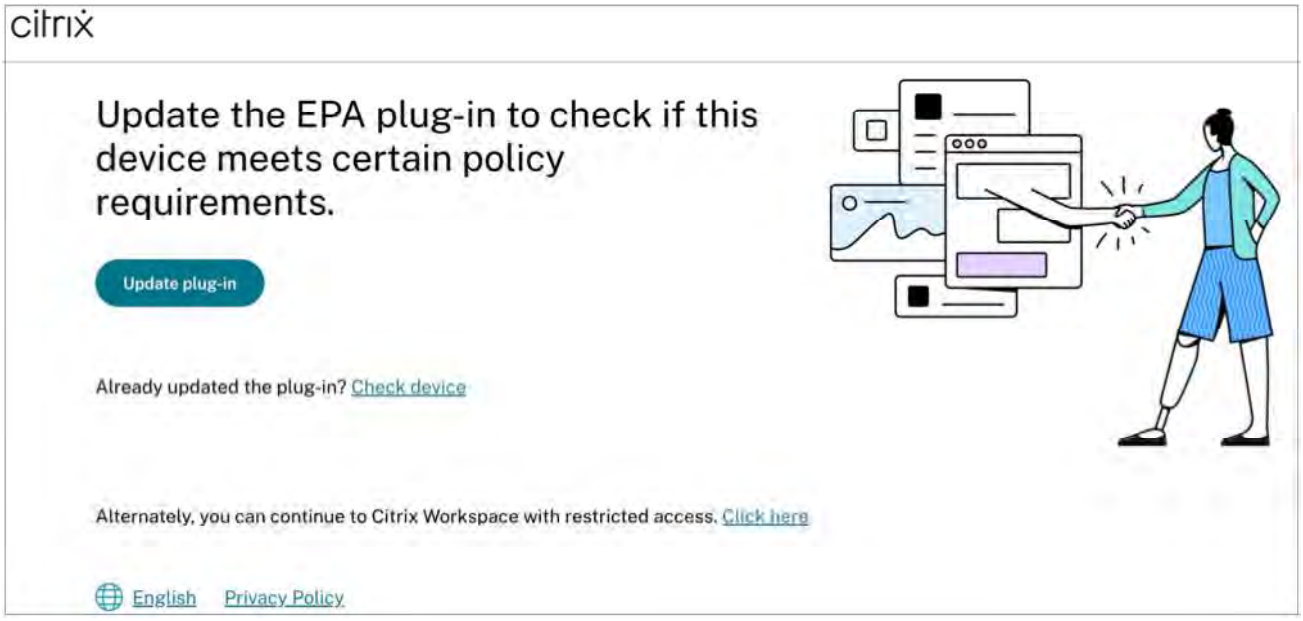
When the skip check feature is enabled, the default policy result (non-compliant) is enforced and the device is classified as non-compliant. End users are provided with partial or restricted access to the Citrix Secure Private Access or Citrix DaaS resources.

Enable skipping the device posture checks

1. Navigate to **Device Posture > Device Scans**.
2. Click **Settings**.
3. In the **Skip device posture check** section, slide the toggle switch ON to enable skipping of the device posture checks.

   When the **Skip device posture check** option is enabled and the end user logs on to the Citrix Workspace, the following message appears when the end user tries to download the client or upgrade the EPA version.

   Alternately, you can continue to Citrix Workspace with restricted access.



Custom workspace URLs support

Custom workspace URLs are supported with the Device Posture service. You can use a URL that you own in addition to your cloud.com URL to access workspace. Ensure that you allow access to citrix.com from your network. For details on custom domains, see **Configure a custom domain**.

Session Recording configuration with Device Posture

Session Recording allows organizations to record on-screen user activity in virtual sessions. You can specify tags when creating a custom session recording policy, event detection policy, or event response policy. For an example, see Create a custom recording policy.

Device certificate check

To configure device certificate checks with the Device Posture service, admins must import an issuer certificate from their device. Once a valid issuer certificate is present in the Device Posture service, admins can use device certificate checks as part of device posture policies.

POINTS TO NOTE:

- Device Posture service supports only PEM issuer certificate type.
- For the device certificate check on Windows, the EPA client on the end device must be installed with administrative rights. For other checks, you do not require the local administrative rights. For details on the supported scans, see Scans supported by device posture.
- To install the EPA client with administrative rights on Windows, run the following command in the location where the EPA client plug-in is downloaded.

  ```
  msiexec /i epasetup.msi
  ```

- The device certificate check with the Device Posture service does not support the certificate revocation check.
- If a device certificate is signed by an intermediate certificate, then you must upload the complete chain containing the root and the intermediate certificates in a single PEM file.

  ```
  Example: chain.pem

  -----BEGIN CERTIFICATE-----
  ******************************
  -----END CERTIFICATE-----
  -----BEGIN CERTIFICATE-----
  ******************************
  -----END CERTIFICATE
  ```

Upload device certificate

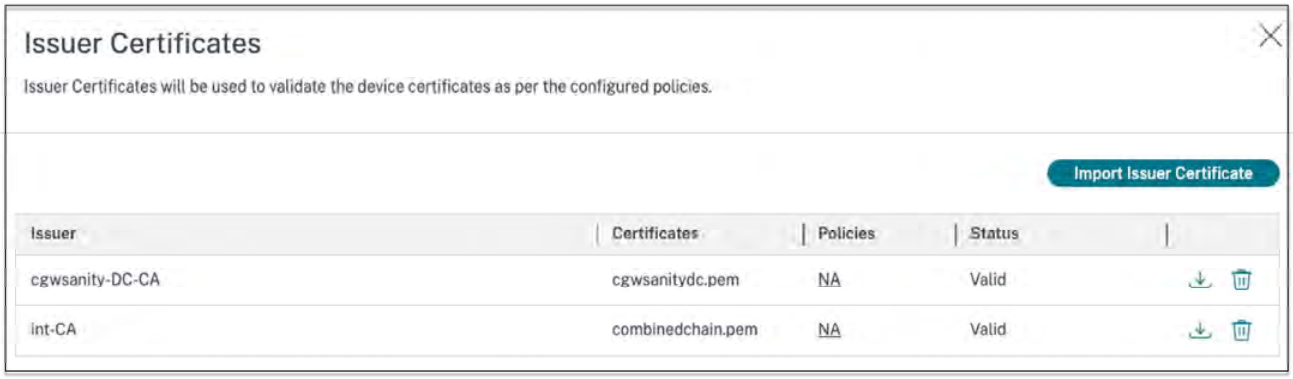
1. Click **Settings** on the Device Posture home page.
2. Click **Manage**, and then click **Import Issue Certificate**.
3. In **Certificate Type**, select the certificate type. Only the PEM type is supported.
4. In **Certificate File**, click **Choose Certificate** to select the issuer certificate.
5. Click **Open**, and then click **Import**.



The selected certificate is listed in **Settings > Issuer Certificates.** You can import multiple certificates.

View imported certificates

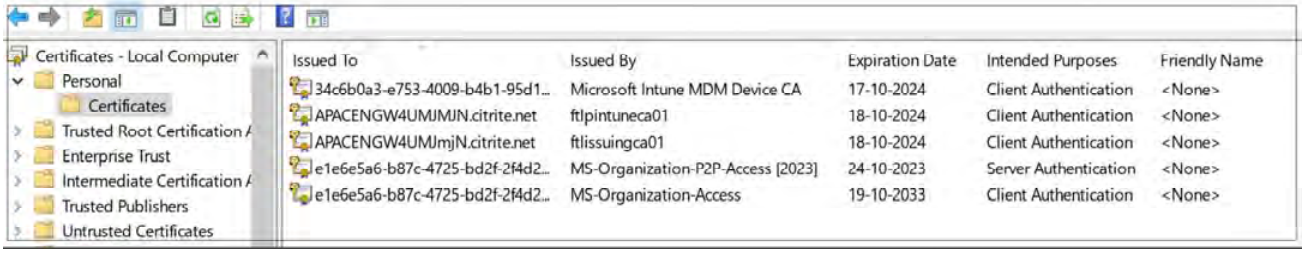
1. Click **Settings** on the Device Posture home page.
2. In **Issuer Certificates**, click **Manage.**
3. The Issuer Certificates page lists the imported issuer certificates.



Install the device certificate on the end device

Windows:

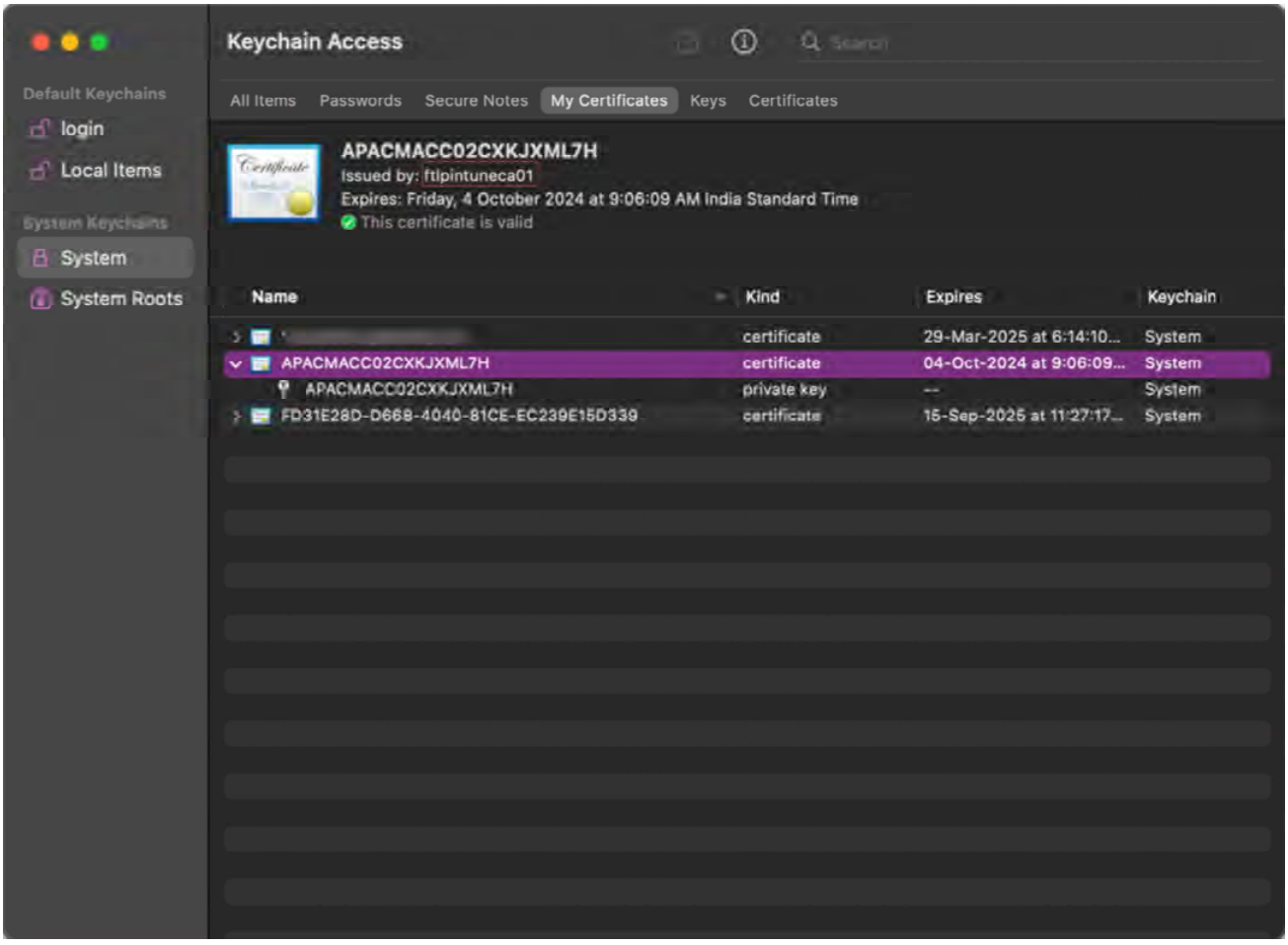
1. From the **Start** menu, open **Computer Certificate manager.**
2. Ensure that the certificate is installed in `Certificates - Local Computer\Personal\Certificates`.
   - The **Intended Purposes** must include **Client Authentication.**
   - The **Issued By** column must match the issuer name configured on the admin GUI.



macOS:

1. Open **Keychain Access** and then select **System**.
2. Click **File > Import items** to import the certificate.

   The **Issued by** field must display the certificate issuer name.



Was this helpful

Send us your feedback

K.Mizra LLC, Ex. 2001, p. 195
Citrix Systems, Inc. et al. v. K.Mizra LLC
IPR2025-01468

Instructions for Contributors

citrix

Site feedback | Your Privacy Choices | Privacy and legal terms | Cookie preferences | docs.cloud.com

© 1999-2025 Cloud Software Group, Inc. All rights reserved.