November 2019

Peer user approval based binary whitelisting

Josh Zukoff

Russell Hancox

Ed Eigerman

Matt Doyle

Matthew Suozzo

See next page for additional authors

Inventor(s)
Josh Zukoff, Russell Hancox, Ed Eigerman, Matt Doyle, Matthew Suozzo, August Huber, and Ben Grooters

Peer user approval based binary whitelisting

ABSTRACT

Enterprises face challenges in monitoring execution of software binaries. This disclosure describes social voting for enterprise level binary whitelisting. Per techniques of this disclosure, a peer user driven approval process is utilized for binary whitelisting. At a time of launch of a binary that is not pre-approved, a user is provided with information associated with the binary and directed to the social voting process. The user designates a peer user and requests that the peer user approve execution of the binary. The peer user is provided with information about the requesting user and about the binary. Approval by the peer user can be used to enable local binary execution by the requesting user. If the peer user does not approve execution, the binary is flagged as blockable, and execution is denied.

KEYWORDS

- Application binary
- Malware
- Blacklisting
- Crowdsourcing
- Social voting
- Executable

BACKGROUND

Enterprises commonly deploy endpoint security systems that are utilized to monitor execution of software binaries (executable code) on enterprise computing devices. Execution of the software binaries is permitted or disallowed based on comparison with a local database of whitelisted (permitted) and blacklisted (blocked/disallowed) binaries. Implementing binary

whitelisting, which requires all software that is permitted to be executed on a machine to be explicitly whitelisted, poses a challenge for large enterprises due to the number of computers and the potentially large number of potential binaries. Large delays may therefore be incurred before system administrators can evaluate each new request for binary execution from users, leading to user frustration and lost productivity.

DESCRIPTION

This disclosure describes the use of social voting for enterprise level binary whitelisting. Per techniques of this disclosure, a peer user driven voting process is utilized for binary whitelisting at the attempted launch of a binary by a user. A check is performed as to whether the binary is currently whitelisted for execution. If the binary is whitelisted, execution is permitted.

If the binary is not currently whitelisted, the execution is blocked, and a peer user driven social voting process utilized to permit local whitelisting and execution of the binary by the user. Fig. 1 illustrates an example workflow for a peer user driven voting process.

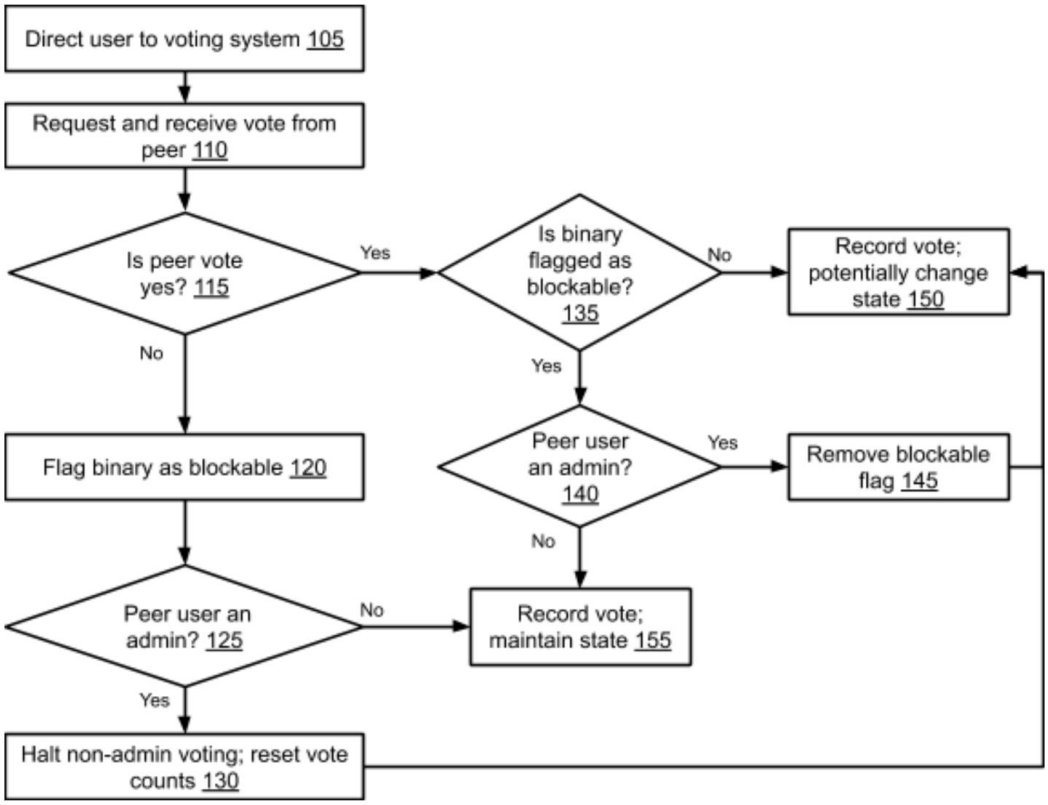

Fig. 1: Example workflow for peer user based binary whitelisting

The user that attempts to launch a binary is provided with information associated with the binary such as metadata, provenance/source of the binary, prevalence of the binary file on enterprise devices, trust signals, signals from one or more anti-virus/malware detection programs, etc. and is directed to the social voting process (105). The user is provided with an option to request a peer to approve execution of the binary.

The user designates a peer user and requests that the peer user approve the binary execution (110). The peer user is provided with information about the requesting user and the information associated with the binary. For example, a simplified summary of the information can be provided to the peer user. The peer user thus has the opportunity to consider the genuineness of the binary application and can utilize the provided information to make a

determination about the binary application. A vote regarding whether the binary may be executed is provided by the peer user.

The vote received from the peer user is evaluated (115). If the peer user does not approve execution of the binary (votes “No”), execution of the binary is blocked (120), else (peer votes “Yes”), flags associated with the binary are evaluated (135).

When execution of the binary is blocked based on a peer vote of “Yes” (115), the binary is flagged as blockable (120), and execution is denied. Next, it is determined whether the peer user is associated with an administrative privilege, e.g., based on peer user role. If the peer user is associated with an administrative privilege, vote counts for the binary are reset, and further voting by users without administrative privilege is halted. Optionally, the state of the binary can be updated (150). If the peer user is not an admin, the vote is recorded and the state of the binary is maintained (155).

If the peer vote is “Yes,” flags associated with the binary are evaluated to determine whether the binary is currently flagged as blockable (135). If the binary is not currently flagged as blockable, the peer vote (approving execution) is recorded, and local execution of the binary by the requesting user is permitted (150). Depending on the configuration, multiple votes of approval by peer users may need to be obtained before execution permission for the binary is provided.

In some configurations, approval by a single peer user is utilized to enable local binary execution by the requesting user and approvals from a predetermined number (multiple) of peer users is utilized to whitelist the binary for execution across the enterprise. This can mitigate systemwide spread of malware by restricting the execution of malware to only a few local devices. If a user obtains approval from a peer user (inadvertently or otherwise) to execute a

piece of software that includes malware, only their local device is thus likely to be affected. Systemwide spread of the malware is blocked since the binary is only locally whitelisted, and additional approval is needed in order for the malware to be executed on other devices.

If it is determined that the binary is currently flagged as blockable, it is determined whether the peer user is associated with an administrative privilege (140). If the peer user is associated with an administrative privilege, the blockable flag previously applied is removed, the binary is permitted to be executed by the requesting user, with the user vote being recorded (150). If the peer user is not associated with an administrative privilege, the user vote is recorded and the state of the binary is maintained (155). Optionally, voting rights can be revoked for users who have repeatedly upvoted malware.

Social voting can be suitable in situations where technology (for example, an automated test) alone cannot determine the safety of a particular binary. The described social voting techniques can also be applied to other blockable events or exception requests. Rather than relying on a small group of individuals, e.g., administrators, to approve a blockable event, or grant an exception, the social voting process can be utilized to seek approval from a group of peer users, thereby reducing the burden on administrators while reducing time spent by the user in waiting for approval.

Recording of user votes is performed with specific permission from users. User can choose not to participate in the social voting process, or participate in a limited way, e.g., request peer approvals, but not provide votes themselves.

CONCLUSION

Enterprises face challenges in monitoring execution of software binaries. This disclosure describes social voting for enterprise level binary whitelisting. Per techniques of this

disclosure, a peer user driven approval process is utilized for binary whitelisting. At a time of launch of a binary that is not pre-approved, a user is provided with information associated with the binary and directed to the social voting process. The user designates a peer user and requests that the peer user approve execution of the binary. The peer user is provided with information about the requesting user and about the binary. Approval by the peer user can be used to enable local binary execution by the requesting user. If the peer user does not approve execution, the binary is flagged as blockable, and execution is denied.