Next-Generation SecureDrop: Protecting Journalists from Malware

Usenix Enigma, January 2020
Jennifer Helsby (@redshiftzero)
Lead Developer
Today

1. Security Goals
2. Current architecture: The story so far
3. Challenges
4. Next generation architecture
5. What’s next
Current Team

100% time on SecureDrop

Jen
Engineering

Kushal
Engineering

Kevin
Support,
Engineering

Allie
Engineering

John
Engineering

Rowen
Support

>= 50% time on SecureDrop

Conor
Engineering

Erik
Project Manager

Mickael
Engineering

Nina
UX

>= 25% time on SecureDrop

Harlo
Training

Olivia
Training

David
Training
A tale of a whistleblower
Ex-CIA officer Jeffrey Sterling jailed for leaking

By Tara McKelvey
BBC News, Alexandria, Virginia

12 May 2015
1. Prevent identification of journalistic sources.

2. Preserve confidentiality of source materials.
CPJ Safety Advisory: Journalist targets of Pegasus spyware

November 6, 2019 11:30 AM ET

The looming threat of newsroom cyber attacks

Recent attacks on the Albuquerque Journal and WBOC reveal the importance of digital security
1. Prevent identification of journalistic sources.

2. Preserve confidentiality of source materials.

3. Prevent journalists from being hacked via malicious submissions.
1. Prevent identification of journalistic sources.

2. Preserve confidentiality of source materials.

3. Prevent journalists from being hacked via malicious submissions.
The story so far...
Some organizations that use SecureDrop for source communication
Current Architecture

Application server: Runs two Python web applications (one for sources, one for journalists) exposed via Tor Onion Services
Current Architecture
Current Architecture

Monitoring server: Runs a host-based IDS (OSSEC) to monitor the application server and send alerts to administrators
Current Architecture

Network firewall: pfSense used to isolate the SecureDrop area of the network from the rest of the news organization.
Documents stored encrypted to the instance’s public key
Current Architecture

Journalists log on using TailsOS
Current Architecture

Journalists log on using TailsOS
Current Architecture
Current Architecture
Current Architecture
Current Architecture

Private key to decrypt documents only in the air-gap environment.
Current Architecture
Accomplishments

1. Minimized the metadata trail between sources and journalists (source traffic is routed through Tor).

2. No third parties to subpoena.

3. If an attacker gets code execution on the workstation with source data, they need to jump the airgap to exfiltrate any data.
Challenges with the airgap
Current Architecture

Cumbersome workflow: Users due to time constraints may circumvent the air-gap.
② **USB drives are reused:** To reduce operational costs and the burden, the same drive is used to traverse the air-gap.
Current Architecture

③ No automatic updates: Significant operational overhead to keeping an air-gapped workstation up to date with security patches.
Malware specifically targeting the air-gap environment: We have seen attacks get code execution and rely on the fact that the submission key is not isolated from the environment in which documents are opened.
Take 2
Technical Goals

1. Ensure known vulnerabilities are patched.
2. Isolate the submission private key from potentially malicious submissions.
3. Isolate each source’s documents.
4. Recover from an attacker getting code execution in the VM used to open submissions.
5. Provide defense in depth to defend against unknown vulnerabilities.
Design considerations

1. Needs to be maintainable by non-specialist IT staff at a news organization.

2. Needs to be usable by journalists.
QubesOS: single-user desktop-based Xen distribution
QubesOS: single-user desktop-based Xen distribution
QubesOS: single-user desktop-based Xen distribution

xen

hardware
QubesOS: single-user desktop-based Xen distribution

Dom0 (fedora)
xen
hardware
QubesOS: single-user desktop-based Xen distribution
QubesOS: single-user desktop-based Xen distribution

Diagram:
- Dom0 (fedora)
- AppVM
- TemplateVM (e.g. Debian)
- xen
- hardware
QubesOS: single-user desktop-based Xen distribution
QubesOS: single-user desktop-based Xen distribution

Only /home, /usr/local, /rw/config will persist through a reboot, otherwise AppVM state is reset to the base TemplateVM.
QubesOS: single-user desktop-based Xen distribution

Upon shutdown, VM is destroyed.

- Dom0 (fedora)
- AppVM
- Disposable VM
- TemplateVM (e.g. Debian)
- Fedora-based
- xen
- hardware
QubesOS: single-user desktop-based Xen distribution

- Dom0 (fedora)
- AppVM
- Disposable VM
- Non-networked VM
- TemplateVM (e.g. Debian)
- Fedora-based
- xen
- hardware
QubesOS: single-user desktop-based Xen distribution

- Dom0 (fedora)
- AppVM
- Disposable VM
- Non-networked VM
- TemplateVM (e.g. Debian)
- Fedora-based

Networking stack runs in sys-net

xen

hardware
QubesOS: single-user desktop-based Xen distribution

- **Dom0 (fedora)**
- **AppVM**
- **Disposable VM**
- **Non-networked VM**
- **TemplateVM (e.g. Debian)**
- **Fedora-based**
- **sys-net**
- **sys-firewall**

Apply firewall rules
QubesOS: single-user desktop-based Xen distribution

USB controllers by default attached here

Dom0 (fedora)

AppVM

Disposible VM

Non-networked VM

sys-net

sys-firewall

sys-usb

TemplateVM (e.g. Debian)

Fedora-based

xen

hardware
QubesOS: single-user desktop-based Xen distribution

Dom0 (fedora)

AppVM

Disposable VM

Non-networked VM

sys-net

sys-firewall

sys-usb

TemplateVM (e.g. Debian)

Fedora-based

xen

hardware
QubesOS: single-user desktop-based Xen distribution

InterVM communication via `qrexec`, based on Xen's `vchan`
Current Architecture
New Architecture
New Architecture

Legend

- qrexec (interVM communication)
- Disposable and non-networked AppVM
- Networked AppVM
- Non-networked AppVM
- System VM
New Architecture

Legend

- qexec (interVM communication)
- Disposable and non-networked AppVM
- Networked AppVM
- Non-networked AppVM
- System VM

sys-net

to internet

sys-firewall
New Architecture

Legend

- qexec (interVM communication)
- Disposable and non-networked AppVM
- Networked AppVM
- Non-networked AppVM
- System VM

sys-net

sys-firewall
tor
New Architecture

Legend
- *qrexec* (interVM communication)
- Disposable and non-networked AppVM
- Networked AppVM
- Non-networked AppVM
- System VM

sys-net

sys-firewall

tor

forwarder

Passes API requests/responses from the SecureDrop server to the user
New Architecture

Legend

- Green: qexec (interVM communication)
- Orange: Disposable and non-networked AppVM
- Green: Networked AppVM
- Yellow: Non-networked AppVM
- Blue: System VM

sys-net

sys-firewall

tor

forwarder

User GUI Application

Non-networked VM used to run a chat-like interface
New Architecture

Legend

- qrexec (interVM communication)
- Disposable and non-networked AppVM
- Networked AppVM
- Non-networked AppVM
- System VM
New Architecture

Legend

- qrexec (interVM communication)
- Disposable and non-networked AppVM
- Networked AppVM
- Non-networked AppVM
- System VM

sys-net

sys-firewall

tor

forwarder

Private key material

User GUI Application

File opening VM

to internet
New Architecture

qrexec (interVM communication)

Legend
- Disposable and non-networked AppVM
- Networked AppVM
- Non-networked AppVM
- System VM

Uses a hardened kernel (grsecurity) in order to provide additional generalized exploit mitigations for memory corruption vulns
Technical Goals

1. Ensure known vulnerabilities are patched.
   • Autoupdates in all VMs (via updating the base templates).

2. Isolate the submission private key from potentially malicious documents.
   • Submission private key is isolated in its own VM.

3. Isolate each source’s documents.
   • Each document is isolated in its own VM.

4. Recover from an attacker getting code execution in the VM used to open submissions.
   • Each file viewing VM is destroyed after shutdown.

5. Provide defense in depth against unknown vulnerabilities.
   • Kernel hardening complicates exploitation of memory corruption-based vulnerabilities.
Journalist Perspective
Current status

1. Audit performed late 2018 of the alpha version of this project (full audit report PDF available on securedrop.org)

2. Beta test beginning in the next few weeks with targeted news organizations

Findings Overview

IncludeSec identified 7 categories of findings. There were 0 deemed a “Critical-Risk,” 0 deemed a “High-Risk,” 0 deemed a “Medium-Risk,” and 5 deemed a “Low-Risk,” which pose some tangible security risk. Additionally, 2 “Informational” level findings were identified that do not immediately pose a security risk.
Takeaways

1. Journalists and their sources face growing challenges due to malware, phishing, and other electronic threats.

2. User-friendly tools for working with potentially malicious documents are critical for journalists.

3. We have built one solution based on QubesOS, but more work in this area is needed.

Interested?

Check out our repositories: https://github.com/freedomofpress/securedrop-workstation

Check out our bug bounty program: https://bugcrowd.com/freedomofpress