ENDGAME.

Dropping Atom Bombs: *Detecting DridexV4 in the Wild*

SERIES ONE, VOLUME ONE

TRAP UNIT

Dropping AtomBombs: Detecting DridexV4 in the Wild

Banking trojans have been around for years, but gained greater visibility in 2015 and into 2016 as they moved from targeting European banks to American banks. We previously discussed the Odinaff banking trojan, which was responsible for the SWIFT attacks, and the theft of close to $1 billion. Another banking trojan commonly known as Dridex steals consumer banking credentials and then makes fraudulent transactions using those credentials. It is making the news due to a major upgrade in capabilities, including the first in-the-wild spotting of the AtomBombing technique. DridexV4 bypasses existing defenses and has already been spotted in campaigns aimed largely at the UK. We'll take a look at DridexV4, the significance of the Atombombing technique, and demonstrate how Endgame's multi-layer approach to prevention and detection proves effective at stopping attacks incorporating never-before-seen techniques.

Background

A new major version of the Dridex malware was recently spotted in the wild. This new version of the Dridex banking trojan marks the first known malicious implementation of the AtomBombing code injection technique, which was first identified in October 2016.

Dridex is a relatively sophisticated banking trojan that has been propagating since 2012. Dridex has been observed targeting online banking customers across Europe and the US, and is one of the dominant banking trojans used by criminals to access banking accounts. Dridex evolved from Cridex, adopting a substantial amount of functionality from the GameOver Zeus banking trojan, which was taken down in 2014 after a global effort led by the US government.

As first reported by IBM X-Force, the new Dridex version implements a novel stealth code injection technique called AtomBombing, which was described in detail in

2016 by researchers at enSilo. AtomBombing is a new code injection technique which doesn't use any of the same API calls executed in traditional code injection, but rather relies on atom tables to deliver code into targeted processes. In general, process injection is a good method to evade defenses because the attacker code hides within a trusted process. API calls used by more common code injection techniques including VirtualAllocEx(), WriteProcessMemory(), and CreateRemoteThread() are often monitored by security products to detect code injection. Avoiding these calls allows the attacker to bypass these security measures and successfully achieve process injection.

DridexV4 in Action

The integration of the AtomBombing technique was a major upgrade for DridexV4, offering the banking trojan even more advanced capabilities to avoid detection. AtomBombing isn't the only interesting technique used by this malware. Upon looking at a DridexV4 sample, we saw a few interesting injection and persistence techniques. The sample utilizes the AtomBombing technique to inject code into explorer.exe, then performs a DLL hijack by copying a random legitimate Windows binary from the "System32" directory into another folder. It then drops a malicious .dll that the legitimate Windows binary loads on runtime. Finally, the sample creates a scheduled task to execute the legitimate Windows file on startup, knowing the malicious .dll will be loaded as well. Additionally, the sample creates a firewall rule that allows the "explorer.exe" process (where the malicious code is injected) to communicate over the network and allow command and control.

The following screenshots depict some of the scripts temporarily dropped to disk and executed to copy system files, add firewall rules, and create scheduled tasks.

Detecting DridexV4

This new version of Dridex is a great example of how adversaries are constantly innovating to defeat the latest security measures. Because of their constantly evolving techniques, a layered approach will always be necessary to consistently detect and prevent attacks from adversaries. In this context, a "layered" approach refers to a multi-faceted solution that cannot be bypassed through any single point of failure. If the adversary bypasses one layer, it will be detected at one of the subsequent layers. Security solutions that do not offer layered behavioral preventions will break down and fail as soon as an attacker finds a creative workaround of the specific defensive technique, which has happened historically time and time again.

Generic Detection

There are a handful of ways to detect and remediate Dridex running in your environment. None of these are guaranteed to detect Dridex, but rather they speak to generic defenses that are required to at least raise the attacker's cost and time for compromise.

- **Audit Autoruns:** Regularly audit all autorun items on all Windows machines in your environment using freely available tools such as Sysinternals Autoruns. Pay especially close attention to any unexpected autorun items that are signed by Microsoft, but not sitting in the same directory as usual.
- **Antivirus:** Ensure you have an anti-malware solution deployed on all endpoints in your environment and the definitions are up-to-date. Antivirus alone is rarely enough to detect a novel variant of malware and detection may be subject to the delays of antivirus signatures.
- **Conduct Memory Analysis:** If you suspect a machine may be compromised, conduct offline memory forensics with an open source tool like Volatility. This can be slow and time consuming but it is one of the few ways to find memory-resident code with open source tools.

Detecting with Endgame

While we offer numerous layers of detection, Endgame detects the latest variants of Dridex in the following places:

- **MalwareScoreTM:** Dridex drops a handful of .DLLs onto disk at runtime that are detected by Endgame's machine learning-driven binary classifier.
- **Fileless Attack Detection:** Once the malicious code is injected into memory with the AtomBombing technique, Endgame's **fileless attack detection** easily locates the malicious code running in memory.
- **Persistence Locations:** The new variants of Dridex drop persistence items as scheduled tasks and DLL side loading. Endgame enumerates every binary that can be executed on startup using all the known techniques from every machine in your environment.

The Only Constant is Change

New offense techniques come and go constantly. Last week, a major new variant of Dridex was observed in the wild, weaponizing the new AtomBombing code injection technique. Security products and organizational procedures that rely heavily on detecting traditional code injection techniques will fail because AtomBombing uses completely different APIs to accomplish an identical end-result of injecting code into memory. A layered approach is the only way to consistently stay on top of the adversaries' newest attack techniques. The new AtomBombing technique is not a one-off abnormality, but is indicative of the sophistication and ever-evolving nature of offensive tradecraft. Just as the attackers pursue multiple and never-before-seen pathways toward successful compromise, so too are multiple layers required to stop them.

ENDGAME.