Exhibit 15

Even for the well aware, with new vulnerabilities discovered every other day, it becomes tedious for a user to go through the ritual of updating the software - closing all applications that use the software, wait for the update to complete and then start all the applications back again.

Anatomy of a Drive-by Download

A drive-by download is a multi-stage attack:

1. The attacker embeds malicious code into an online advertisement displayed on a trusted website.
2. A user visiting the website gets redirected to the attacker's site without the user clicking on the advertisement.
3. An exploit kit from the attacker's site looks for possible vulnerabilities on the user's endpoint.
4. Based on the exploit discovered, a desired malware is downloaded to the endpoint without the user's knowledge.

A drive-by download is a sneaky attack where a user normally browsing a seemingly harmless site can get infected without clicking on anything. The benign website can be compromised in different ways - by embedding malicious code in a comment field on a blog or a poorly secured web form. But the easiest way to go about this is by taking advantage of a flaw in an online advertisement and injecting malicious code in it. Trusted websites that are visited by thousands every day can end up hosting advertisements running malicious code without their knowledge.

The malicious code injected into the advertisement redirects the user to the attacker's website by loading the malicious url in a new window. This new window goes undetected because attackers make use of a common HTML feature called Inline Frame or iFrame for short. An iFrame is an HTML document that is embedded into another HTML document. For example, a YouTube video can be seamlessly embedded into a main webpage. In reality, it is just a regular webpage playing a YouTube video that is inserted into the main page by adjusting the size and removing the borders, it gives an illusion that the YouTube video is actually a part of the main webpage. So when the malicious code redirects the user to a different website, it opens up in a tiny window which can't be easily spotted by the human eye.

Once the user gets redirected to the attacker's web page, an exploit kit examines the endpoint for possible vulnerabilities to take advantage of. This is the beginning of the attack. The exploit kit gathers information about the operating system, browser type, browser version and browser plugins and looks for security holes in them. Browser plugins such as Java Runtime Environment, Adobe Flash Player, Adobe Reader are popular targets. The exploit itself doesn't cause any actual damage - the security codes of the building have been cracked, but nothing has been stolen yet.

Armed with the knowledge of how to attack the victim, the exploit kit proceeds to download an appropriate malware to the victim's endpoint. The malware also known as "payload" is automatically installed on the endpoint without the user's knowledge. The payload download goes unnoticed because it is usually obfuscated. Obfuscation is a common technique used by attackers to evade traditional signature based detection engines and helps mask the real purpose of the malicious code. Once the malware has been downloaded and executed, it proceeds to do what it's best designed for - to make some green for the attacker. The malware can extract crucial banking information or lock your folders in exchange for money (more commonly known as Ransomware). Even more insidious attacks may start with reconnaissance tools that stay "low and slow" and take stock of critical assets on the network and sniff for access credentials.

Each of these existing solutions try to convince users that their "silver bullet" will protect users from drive-by downloads but adding more functionality or signatures to products that were originally designed to detect viruses or malicious websites, is no match for the sophisticated attacks we have seen in the past few years.

The Cyphort Solution

Cyphort has been designed from the beginning to address the dynamic nature of Advanced Persistent Threats. For a complicated problem such as a drive-by download, a single, traditional approach will not do the trick. Cyphort attacks the problem from different angles:

- **Chain Heuristics:** Cyphort uses a heuristics model to identify potentially malicious traffic. As there are thousands of web pages being visited by employees in a company, this is a crucial step to focus on interesting traffic and provide quick results. Cyphort analyzes all the traffic and looks for some indicators such as "Is this browser running a vulnerable version of a browser plugin", "Was this web page referred from a valid resource link", "Why is a field missing in the header", "Is this webpage part of a trusted domain" and other such questions.
- **Browser Behavior Analysis Engine:** If a particular HTTP session is determined to be potentially malicious by the heuristics model, more analysis is done to confirm the verdict. The entire HTTP session is simulated using a browser that runs in Cyphort's sandbox environment. Cyphort examines the browser logs and downloaded artifacts to confirm any suspicious activity.
- **Dropper Analysis:** Cyphort looks for any executable artifacts (dropper) that are downloaded as part of the chain. Cyphort subjects the dropper to static analysis, behavior analysis and reputation analysis to identify if it is a malware.

Cyphort's true strength in combatting drive-by downloads lies in using a combination of techniques to counter different kinds of exploits. Each exploit has its own traits and it would be difficult to detect them all with a single method approach.

Every exploit has a tell - but it is important to know what to look for or it could easily end up being a wild goose chase. These clues are subtle and spread across several requests and responses. Chain Heuristics does not look at packets as mere zeroes and ones that it can match a signature against, it understands the context by inspecting the sequence of HTTP requests and responses between a particular source and a destination. Each of these sequences is called a chain. Chain Heuristics checks for suspicious indicators in the headers and body of each HTTP request and response and also overall in each chain.

The suspicious indicators get constantly updated depending on what exploits are out there. Cyphort Labs researchers study new exploits in the wild and come up with these indicators. The indicators by themselves may not draw attention, but when all the indicators are added up along with enough context, things will start to look suspicious. For example, consider an endpoint in an enterprise that fetches a few webpages from an outside web server hosted on port 8000. That doesn't seem suspicious at all. A lot of web servers run on non-standard ports for enhanced security, but if the same endpoint also downloads an encrypted executable file and its browser runs a vulnerable version of a browser plugin, then things begin to fall into perspective. The strength of Chain Heuristics lies in the context that is extracted from the traffic. With threat intelligence data from Cyphort's Malware Researchers combined with Heuristics, this solution offers a unique angle to the problem.

Depending on the verdict obtained from Chain Heuristics, Cyphort decides if the suspicious chain needs to be looked at by the Browser Behavior Analysis Engine. It recreates the attack by executing