# Global Information Assurance Certification Paper

# Smaller Devices Need a Large Firewall.

*GIAC (GCIH) Gold Certification*

Author: Paul Mastad, paulmastad@outlook.com
Advisor: Manuel Humberto Santander Peláez

Accepted: August 4th 2009
(Date your final draft is accepted by your advisor)

Abstract

The term “smaller devices” refers to a group of devices comprised of both tablets and smartphones. Palo Alto Networks next-generation firewall (PAN) is normally used to protect corporate networks. PAN can also protect smaller devices, and features like WildFire can combat malicious Android apps. This paper examines WildFire in detail. PAN’s RESTful XML API, together with PowerShell, allows for flexible management.

# 1. Introduction

Palo Alto Networks (PAN) next-generation firewall encapsulates a full line of products. The highest-end PAN platform is the chassis-based PA-7050, while mid-range models include the PA-3000 and PA-2000 series. The PA-200 is the low-end model. There is also a similar range of virtual platforms ranging from the VM-1000-HV, a high-end virtual model, to mid-range models like the VM-200. The VM-100 is the lowest-end virtual platform. PAN low-end models have less throughput than the high-end Enterprise models, but, in spite of this, retain almost all of their functionality and capabilities.

PAN offers a set of add-on features. App-ID is a traffic-classification system designed to identify applications. It does this by utilizing ports, signatures, and transmission characteristics. Content-ID is similar to traditional anti-virus signatures. This group includes URL filtering, data filtering, and IPS (Intrusion Prevention System). User-ID integrates PAN into directory services. PAN uses User-ID to grant access to users instead of IP addresses. Finally, Panorama is a management product which is purchased separately. A central Panorama can administer multiple PANs.

GlobalProtect is a Palo Alto Networks add-on family designed to extend the coverage of a PAN unit. Traffic from enrolled devices uses a VPN to reach a PAN, and PAN may route the traffic internally or towards the Internet. Traffic from outside devices is then subject to the same policies and protection as inside devices. Devices with GlobalProtect will show the IP of the PAN. A central GlobalProtect may protect smaller devices owned by the corporation and may additionally safeguard BYOD (Bring Your Own Device) users.

Earlier versions of PAN used only what they termed “single-pass parallel processing (SP3) architecture” (Palo Alto Networks, 2011). The term “single-pass” refers to the software component of PAN. “Parallel processing,” describes the hardware. With the introduction of virtual PANs, however, this term cannot be said to accurately apply to all models. Despite this, it still offers a quick explanation of some operating principles. Essentially, PAN software handles packets only once. The “parallel processing” hardware, however, performs multiple operations at one time. Examples

Paul Mastad paulmastad@outlook.com

include routing, policy lookup, threat prevention, and a number of miscellaneous networking tasks. The intent is to avoid latency while at the same time quickly stopping malicious traffic.

WildFire is a Palo Alto Networks add-on product family with an automated sandbox test environment at its core. It has a different approach than the "single-pass parallel processing (SP3) architecture." WildFire will buffer files. Interestingly, it will not stop the first occurrence of new malware. Instead, when WildFire discovers new malware, it will do two things. First, it will pass on a warning to the reporting functions of PAN. Then, it will contribute to blocking the next occurrence of this particular malware. PANs with WildFire may be set up to cooperate, and cooperating PANs will share information in order to create WildFire virus signatures.

The term "smaller devices" refers to a group of devices comprised of both tablets and smartphones (International Data Corporation, 2013). Palo Alto Networks next-generation firewall (PAN) can protect smaller devices. This paper will focus on PAN's ability to discover unknown, malicious Android apps using a working, real-life setup. The scenario used will be the borderline between corporate and personal devices. PAN's RESTful XML API (PAN API) programming interface, together with PowerShell, constitutes an alternative way to administer PAN devices. Three scenarios will demonstrate PAN API and PowerShell.

## 2. A new environment for PAN.

Until now, PAN has mainly been used by governments, corporations, and other structured organizations. Maintaining a PAN device requires specific skills related to firewalls and networking as well as insight into how solutions and applications behave on a corporate network. Reusing the same skillsets may help to protect smaller devices against malicious apps.

Smaller devices contribute to the blurring of the work-life distinction. Some mobile users will occasionally end up completing work-related tasks on their private devices over their home WiFi. GlobalProtect VPN tunnels to corporate PANs may not work in all situations. One example of this is the restrictions employed by the TV

distribution company GET. In addition to providing TV, GET is also an Internet Service Provider and offers an app to watch TV on smaller devices. The app only works with an IP address provided by GET. GlobalProtect and its VPN tunnel to a corporate PAN will present a different IP address. Consequently, GET's TV app will not work with a centralized GlobalProtect.

An alternative to a centralized GlobalProtect is to deploy several low-end PANs. Risk analysis is useful to determine whether remote workers should have low-end PANs on their home WiFi networks. IT staff can be used to beta test the deployment of low-end PANs. One factor that should be taken into consideration is that home networks are likely to be used for non-work related activities. In addition, an entire household may use the home network. As a consequence of this, PANs on home networks generally require a softer approach with less dogmatic rules.

PANs are expensive. As such, it is important to use care in selecting only important WiFi networks upon which to deploy a PAN. The corporation's internal WiFi and guest WiFi networks should be added to existing PANs. This approach has two advantages: the existing PANs have already been purchased, and the supporting organization is in place. Existing PANs are good initial candidates for the protection of smaller devices. It is important to reuse the experience gained from existing PANs when deploying low-end PANs.

## 2.1. Circumstances where PAN is out of place.

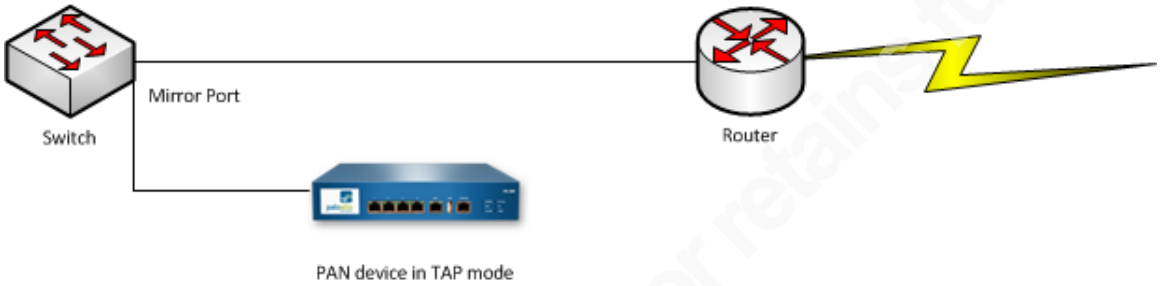
If risk analysis indicates that smaller devices need protection, PAN may be an option. In the planning stage, both budget and privacy should be addressed. PAN devices are costly; personnel with PAN skills are necessary for successful operation, and these personnel are typically expensive. If the budget is insufficient, alternatives should be considered.

PAN normally collects a considerable amount of information about the traffic that passes through it. It has advanced reporting capabilities, and these reporting capabilities can be used or misused. For home networks, clarify who owns the logs collected by the PAN and define the usage of the logs in advance. The legal term "privacy" may have

different definitions in different countries. Legal assistance and counsel regarding privacy laws should be obtained before deploying PANs on any home network.

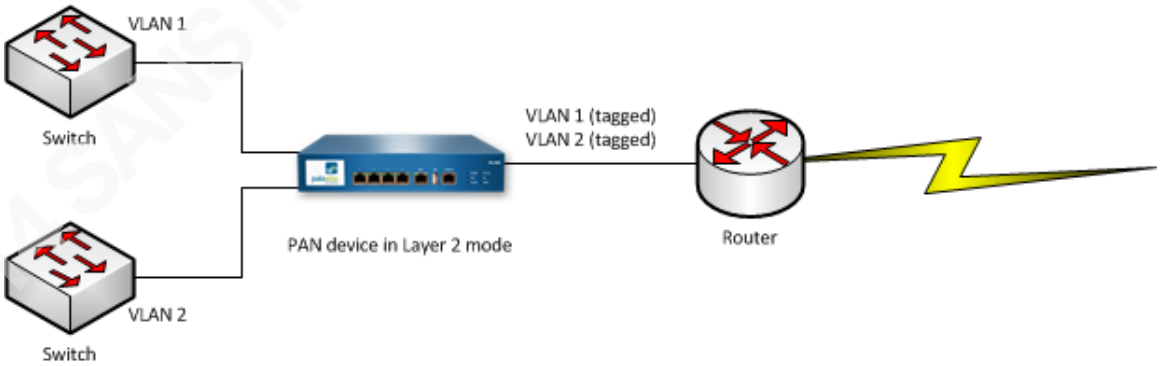
## 2.2. Minimum interruption mode.

PAN has four possible modes. In TAP deployments, a switch copies the traffic. PAN receives a copy of the traffic. It transmits no traffic itself; it is a passive observer in TAP deployments. The other three modes allow PAN to react to traffic.



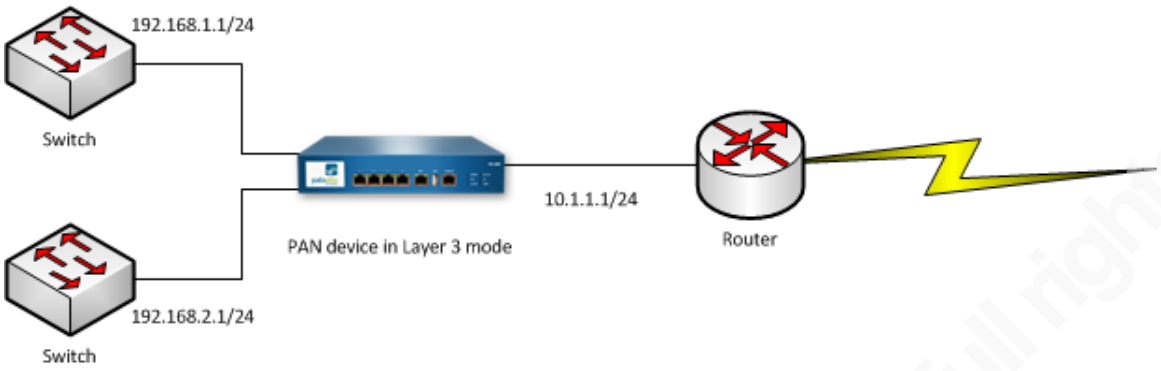

Virtual Wire mode makes PAN a transparent Ethernet connection and provides neither switching nor routing inside the PAN.


The third approach is Layer 2, in which the PAN switches traffic between two or more networks.




Layer 3 is the fourth method, which routes traffic between multiple PAN ports.

Paul Mastad paulmastad@outlook.com




In this paper, the PAN device is set to Virtual Wire mode. The Wi-Fi access point and broadband router are two separate devices, which is a pre-requirement for Virtual Wire mode. Virtual Wire requires few if any changes to the neighboring devices. There are no changes to IP or VLAN settings. For part of the time, Virtual Wire behaves similarly to a transparent bridge. It binds together the Ethernet ports on the Wi-Fi access point and broadband router. These two devices see and use each other's MAC addresses. In normal operation, the PAN device is transparent. Blocking PAN policies will, when triggered, halt offending traffic streams.


In theory, Virtual Wire should be a plug-and-play configuration. In real life, this is not always true. Virtual Wire mode PAN and the existing equipment may need some tinkering before they play well together.

In this particular example, there was no need to change the configuration of the WiFi device or the broadband router. The device PAN-LAB in this paper had both its Virtual Wire interfaces set to default, which is Link Speed - auto, Link Duplex - auto, and Link State - auto. Both interfaces had to be manually adjusted. The modified settings were Link Speed - 100, Link Duplex - full, Link State - up. Finally, all three devices (WiFi, PAN-LAB, and Broadband Router) had to be powered OFF and then ON. Only then did this particular setup work.

Paul Mastad paulmastad@outlook.com

## 2.3. Automation using WildFire.

WildFire was first included in version 4.1 on November 2011. Subscription services came one year later with version 5.0. Wildfire initially supported PE files (Portable Executables) or MS Win32 executables, commonly known as .exe and .dll files. "Specimen 1" is an example of a typical malicious .exe file (the e-mail warning can be found in Appendix A). Specimen 1 carried out a series of suspicious activities, such as "Created an executable file in a user document folder" and "Modified registries or system configuration to enable auto start capability."

PAN Version 6.0 was released in January of 2014. New WildFire features in Version 6.0 include support for additional file types. Additionally, support was added for Java JAR, Android APK, PDF, and MS Office Docs files. Different licensing options include or exclude WildFire features depending on the option chosen. This paper will use the full set of WildFire features for version 6.0.

"Specimen 2" is a free android app downloaded from Google's Play Store. It illustrates what an app may do and still be deemed benign. Specimen 2 is used later in the paper to demonstrate how to download only selected parts of the report. Appendix B contains the PDF report for Specimen 2. The details reveal some interesting information in this benign file. Specimen 2 uses "Suspicious API Calls," according to the WildFire report. This app stated, before installation, that it needed "Full network access." In this case, usage of the API "android/net/ConnectivityManager ;-> getActiveNetworkInfo" is no surprise. DNS queries show that Specimen 2 is using Google Analytics. It is presumably collecting usage and traffic statistics on behalf of the creators of the app. The payment for this "free app" is privacy.

### 2.3.1. Verdict, malicious or benign.

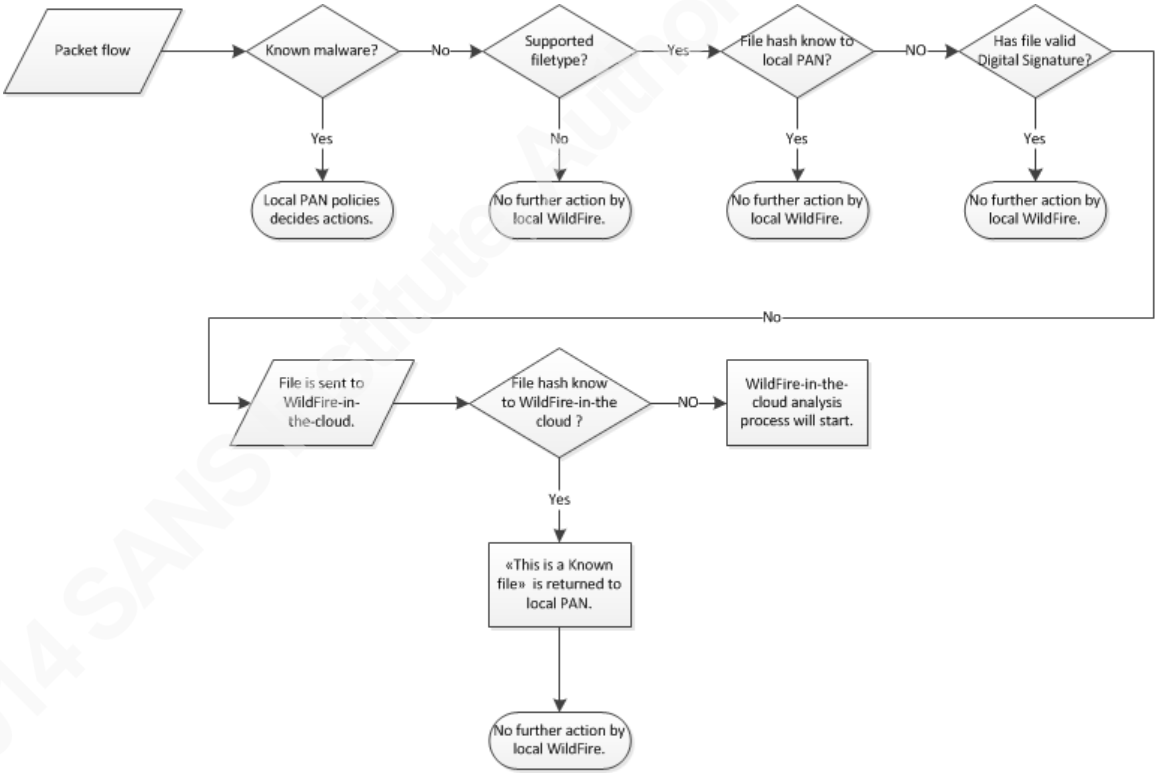
The exact verdict process in WildFire is a trade secret; however, a comparison between the Funtasy Trojan and Specimen 2 gives us some insight into its workings. The Funtasy Trojan was uncovered and studied by a team from Palo Alto Networks. Specimen 2 is a benign app reviewed by the PAN device used in this paper.

The Funtasy Trojan requires extensive permissions, which it aggressively misuses once installed. The Trojan will edit and rearrange SMS messages in order to conceal

illegitimate charges being made to the user (Palo Alto Networks, 2014). Specimen 2 has access to and uses "Suspicious API Calls." Specimen 2 makes use of those API Calls in a more polite way; it generates data traffic and is open about it. The sum of the observed behaviors is the difference between a malicious and a benign app.

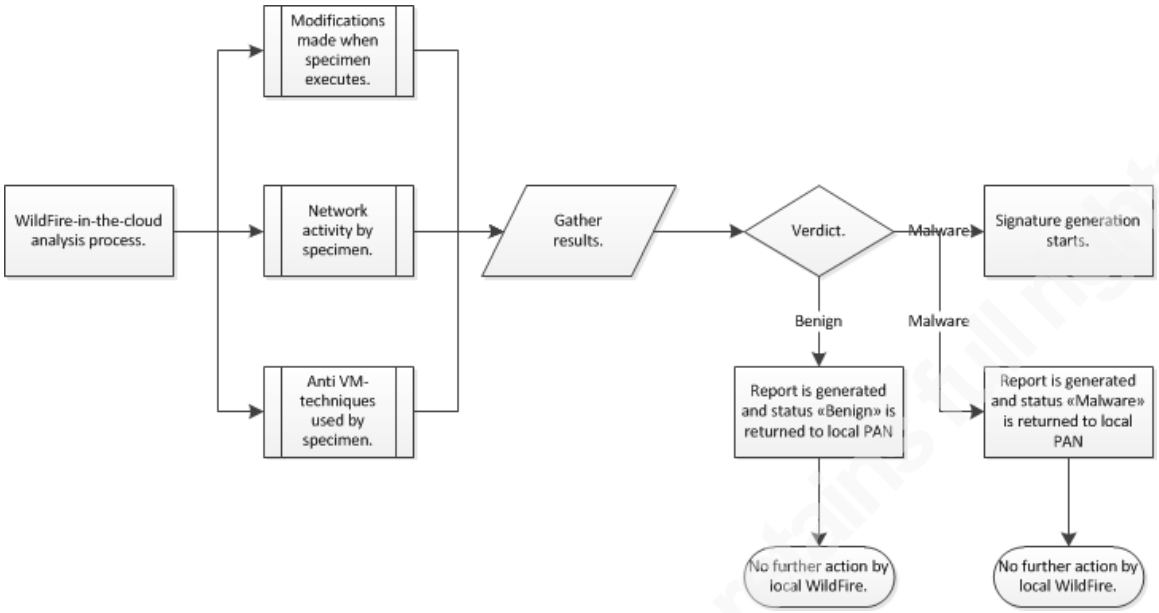
### 2.3.2. The WildFire Process.

The WildFire process starts when traffic triggers a file-blocking rule. This rule must have a "forward to WildFire" option. Supported file types are scanned, and, if presumed OK, buffered and hashed; then, the file is checked to determine if it has been previously identified. WildFire uses the SHA-256 algorithm by default. Files that have been found earlier are not checked again. Valid digital signatures exempt files from further checks.

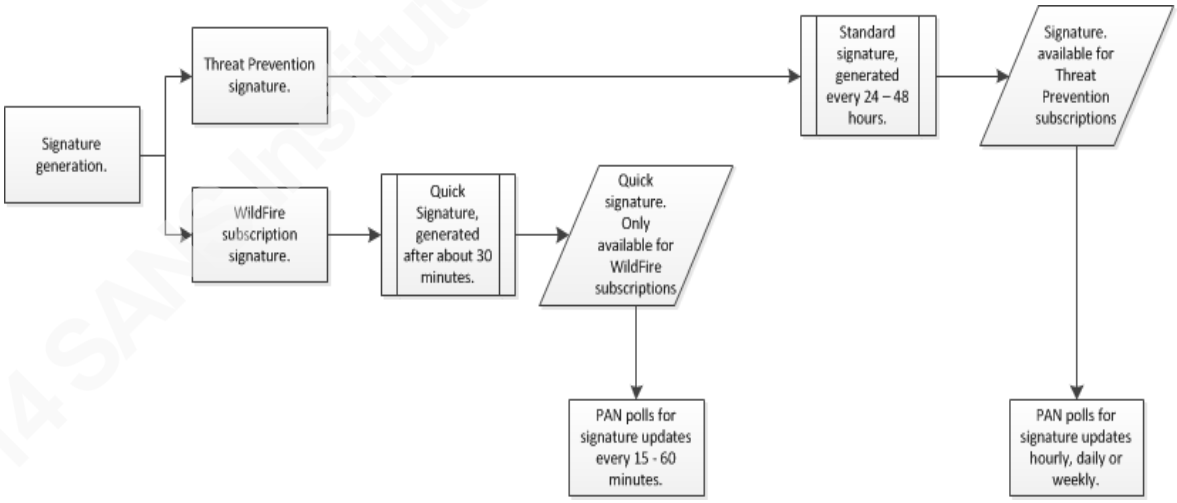



WildFire-in-the-cloud receives unknown files for further analysis. Sandboxing is used to execute the unidentified file in a controlled environment. Its behavior is observed before, during and after the unidentified file is executed. What the file does when executed determines the verdict. A benign file will be reported as such and becomes known as an identified file.

Paul Mastad paulmastad@outlook.com




A file deemed malicious triggers further actions. An automatic process creates a new signature. This new signature will detect the newly found malicious code. PAN devices with WildFire subscriptions can download updated signatures within one to two hours. PAN devices without the WildFire subscription can download an updated signature within 24 to 48 hours.




### 2.3.3. Limits to WildFire.

WildFire is an automated process, and it has its limits. For one, it is a closed and confidential process. The documentation provided with WildFire generally skips the specifics of how it works. Observation, however, does reveal some details. Files signed by Microsoft may be checked, as they were unsigned. Office 2010 on-demand and

WSUS may transfer files in packages, and PAN may fail to recognize such packages as signed files[1]. The WildFire process may then waste time checking those files. WildFire analyzes behavior, not code. Event X may trigger a particular malicious code. If event X is missing, the malicious code will pass undetected. Moreover, there is a Virtual Machine (VM) arms race. There are ways to detect if an application is running on a VM. In addition, there are ways to camouflage the VM as a physical device.

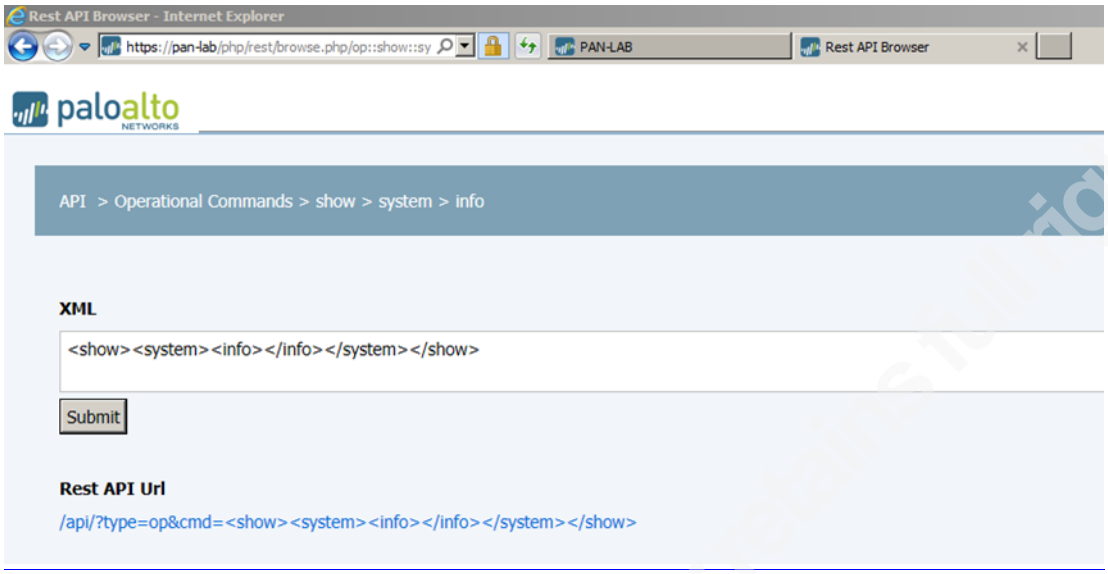
## 2.4. PAN OS XML-based REST API

PAN has several means of administration. Web GUI or CLI may be chosen based on personal preference or the task that is to be performed. A central Panorama can administer multiple PANs. PAN's RESTful XML API (PAN API) can be either an alternative or a supplement to Panorama, as PAN API has features that can manage multiple PANs more flexibly than Panorama.

This paper focuses on Microsoft PowerShell version 4 together with PAN API. Integrated PowerShell was provided with Windows Server 2008 R2. PowerShell is available for all supported versions of Microsoft Windows. PowerShell has two traits that make it useful for administrators. As a one-liner, it can accomplish a specific task from CLI. Furthermore, it is a powerful scripting language. Invoke-RestMethod is a native PowerShell command. PowerShell has no need for third-party tools to interact with PAN API.

PAN API's Rest API Browser is a good starting point. It gives insight into the structure of the PAN API. It is also capable of issuing one-line commands. Access to the Rest API Browser is similar to the web-based GUI. A URL such as https://pan-lab allows login to the Web GUI. When the user has logged in, he or she may modify the URL or open an additional tab in the web browser: https://pan-lab/api.

[1] Observed behavior – https://www.virustotal.com/en/file/2ce2ee9fdcc263bd5e9afe79d1cc93025b5c8d0ef8c9c3fc5912cd08cfe0c743/analysis/



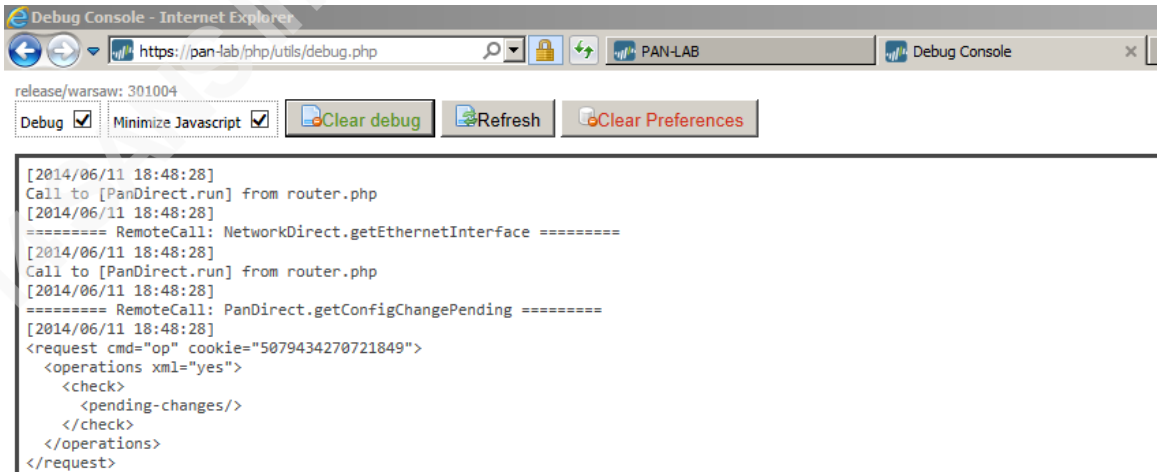

Another way to find the commands is to use the CLI in debug mode.

admin@PAN-LAB> **debug cli on**

admin@PAN-LAB> **show system info**

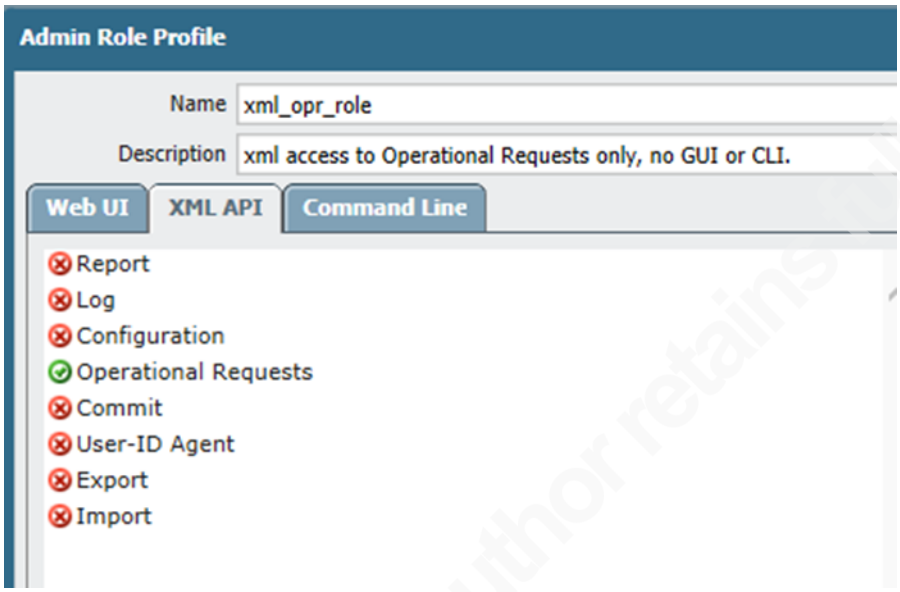
https://pan-lab/api/?type=op&cmd=%3Cshow%3E%3Csystem%3E%3Cinfo%3E%3C%2Finfo%3E%3C%2Fsystem%3E%3C%2Fshow%3E&key=987654321

Debug is also available from GUI. : https://pan-lab/debug




The Rest API Browser has eight main categories. If desired, an admin user may be granted access to only one of the categories. To accomplish this, it is necessary to use

the GUI's "Device - Admin Roles" to create an XML role with no privileges beyond those necessary. Then, one only has to create a new custom role-based administrator in "Device - Administrators."




PAN API uses an authentication key, which is generated when a username and password is entered (see section 3 for examples). PAN does not use salting; however, it does use the concept of master keys (see GUI's "Device - Master Key and Diagnostics"). Some scenarios may benefit from having the same key for multiple PAN units. In other scenarios, having one key to unlock them all may be a liability.

WildFire-in-the-cloud and WildFire Portal have their own RESTful API interface. The WildFire API Programming Guide is only accessible after logging in to the support pages.

## 3. 0 Scripting with PAN's API.

Here, three scenarios will be considered in order to demonstrate the usage of PowerShell together with PAN API. The three scenarios are a routine check, closer examination of a suspicious event, and adjustment of the configuration.

Paul Mastad paulmastad@outlook.com

## 3.1. Routine check

When managing multiple PAN devices, uptime is a good indicator of the system's health. An uptime which appears to be too low indicates unplanned reboots. A PAN system that has been running too long may become unstable. User ID Xml_opr_user has access only to XML Operational Commands.

# Retrieves and stores the key phrase. It is stored in memory for later use.
PS_C:\>**$keyOPR=Invoke-RestMethod -Uri "https://pan-lab/api/?type=keygen&user=xml_opr_user&password=Answer2EverythingOPR"**
PS C:\ > **$keyOPR=$keyRO.response.result.key**

# Collect the result of "show system info" and stores it in memory.
PS_C:\>**$pan=Invoke-RestMethod -Uri "https://pan-lab/api/?type=op&cmd=%3Cshow%3E%3Csystem%3E%3Cinfo%3E%3C%2Finfo%3E%3C%2Fsystem%3E%3C%2Fshow%3E&key=$keyOPR "**

# Removed the key phrase from memory.
PS_C:\>**$keyOPR="none"**

# Alternative 1) – Display the uptime for manual control.
PS_C:\>**$pan.response.result.system | Format-List -Property devicename,uptime**
devicename : PAN-LAB
uptime: 15 days, 8:01:50

# Alternative 2) – Compare the uptime to a desired value.
#Checks if uptime is greater than the limit. If so, then it writes a notification.
PS_C:\>**$uptime=$pan.response.result.system.uptime**
PS_C:\>**$uptime=$uptime.Substring(0,3)**
PS_C:\>**$uptime=$uptime.trim()**
PS_C:\>**$uptimeint=[int]$uptime**
PS_C:\>**IF ($uptimeint -gt 15) {"Uptime is more than 15 days"}**

Paul Mastad paulmastad@outlook.com

# Alternative 3) – Write to a log file or to transfer the result to a management system.
PS C:**$pan.response.result.system | select-object devicename,uptime | convertto-csv -NoTypeInformation | % { $_ -replace ‘“‘, ‘“‘} | out-file .\panUptime.csv**
PS_C:\>**type .\panUptime.csv**
devicename,uptime
PAN-LAB, 15 days, 8:01:50

## 3.2. Closer examination of a suspicious event

This script retrieves the WildFire logs for the past day. It then goes on to pull a detailed report from the WildFire cloud (see report on Specimen 2 in Appendix B). The first portion of the script uses the PAN API. WildFire-in-the-cloud has its own API. The second part uses WildFire API. User ID Xml_log_user has only access to XML Operational Commands.

# Retrieves and stores the key phrase. It is stored in memory for later use.
PS_C:\>**$keyLOG=Invoke-RestMethod -Uri “https://pan-lab/api/?type=keygen&user=xml_log_user&password=Answer2EverythingLOG”**
PS_C:\>**$keyLOG=$keyLOG.response.result.key**

**#** Stores the date in memory for later use.
PS_C:\>**$date=((get-date).AddDays(-1)).ToString(“yyyy/MM/dd HH:mm”)**

# Performs the query.
PS_C:\>**$query2=Invoke-RestMethod -Uri “https://pan-lab/api/?type=log&log-type=wildfire&query=( receive_time geq ‘$date’)&key=$keyLOG”**
PS_C:\>**$jobid=$query2.response.result.job**

# Checks if the log search job is done.
PS_C:\>**$jobStatus=Invoke-RestMethod -Uri “https://pan-lab/api/?type=log&action=get&job-id=$jobid&key=$keyLOG”**

Paul Mastad paulmastad@outlook.com

PS_C:\>**$jobStatus.response.result.job.status**

FIN

# Retrieve the query result only if the query job is completed.

PS_C:\>**$query2=Invoke-RestMethod -Uri "https://pan-lab/api/?type=log&action=get&job-id=$jobid&key=$ keyLOG "**.

# Verdict in this case was benign.

PS_C:\>**$query2.response.result.log.logs.entry.category**

Benign

# Removes the query job.

PS_C:\>**$deljob=Invoke-RestMethod -Uri "https://pan-lab/api/?type=log&action=finish&job-id=$jobid&key=$ keyLOG "**

# Removed the key phrase from memory.

PS C:\>**$keyLOG ="none"**

# Filedigest is the term used in this PAN log for SHA-256 checksum.

# Filedigest is the unique identifier used against WildFire-in-the-cloud.

PS C:\>**$query2.response.result.log.logs.entry.filedigest**

5f12ea5db95fb9a8abbb0c16e2af7c644d5f9ffdc2495358fd8fe4ae3c70ff60

# The WildFire-in-the-cloud API key is stored in memory for later use.

PS_C:\>**$apikey=a870c11111111111793509000000000**

# SHA-256 / Filedigest is stored in memory for later use.

PS_C:\>**$SHA-256=$query2.response.result.log.logs.entry.filedigest**

# The input file may be found in Appendix C.

PS_C:\>**$body = "$(get-content D:\source\input.txt -raw)"**

PS_C:\>**$body = $body.Replace("ReplaceThisStringWithSHA-256", "$SHA-256")**

Paul Mastad paulmastad@outlook.com

PS_C:\>**$body = $body.Replace("ReplaceThisStringWithApikey", "$apikey")**
PS_C:\>**$ContentType = "multipart/form-data; boundary=----------0a8d842719f1707d"**

# Requests the report for specific event identified with SHA-256.
PS_C:\>**$report2 =Invoke-WebRequest "https://wildfire.paloaltonetworks.com/get-report-xml" -ContentType $ContentType -Method Post -body $body**
# $report2 – see the report for Specimen 2 in Appendix B.

# As an alternative to reading the entire report, selected key areas may be pulled.
PS_C:\>**$sensitive_API= $report2| Select-Xml -XPath "//Sensitive_API_Calls_Performed"**
**PS_C:\>$Sensitive_API.Node.entry**

android/net/ConnectivityManager;->getNetworkInfo
android/telephony/TelephonyManager;->getDeviceId
android/accounts/AccountManager;->getAccounts
android/net/wifi/WifiManager;->getConnectionInfo
android/net/ConnectivityManager;->getActiveNetworkInfo
android/webkit/GeolocationPermissions$Callback;->invok
android/app/NotificationManager;->notify

PS_C:\>**$defined_sensors= $report2| Select-Xml -XPath "//Defined_Sensors"**
PS_C:\>**$defined_sensors.Node.entry**

Receive sensor readings from gps.

## 3.3. Adjusting the configuration

PAN API can change the configuration of a PAN. This example will adjust the anti-virus profile Antivirus-log-only. First, find what is to be changed and where it is located in the XML configuration. "Show running config" from within the Rest API Browser is a good start. API > Operational Commands > show > config > running.

Paul Mastad paulmastad@outlook.com


# Retrieves and stores the key phrase. It is stored in memory for later use.

PS_C:\>**$key=Invoke-RestMethod -Uri "https://pan-lab/api/?type=keygen&user=xml_user&password=LifeUniverse42"**

PS_C:\>**$key=$key.response.result.key**

# Defines the path. It is stored in memory for later use.

.PS_C:\>**$xpath="%2Fconfig%2Fdevices%2Fentry%2Fvsys%2Fentry[@name='vs**

Paul Mastad paulmastad@outlook.com

**ys1']%2Fprofiles %2Fvirus%2Fentry[@name='Antivirus-log-only']**
**%2Fdecoder%2Fentry[@name='smtp']"**

# Defines the element. It is stored in memory for later use.
PS_C:\>**$element="%3Centryname='smtp'%3E%3Caction%3Eblock%3C%2Facti**
**on%3E%3Cwildfire-action%3Eblock%3C%2Fwildfire-action%3E%3C%2F**
**entry%3E"**

# Sends the changes.
PS_C:\>**$pan=Invoke-RestMethod -Uri "https://pan-**
**lab/api/?type=config&action=edit&key=$key&xpath=$xpath&element=$element"**
PS_C:\>**$pan.response.msg**
command succeeded

# A partial commit saves time. It updates only the chosen section.
PS_C:\>**$pan=Invoke-RestMethod -Uri https://pan-**
**lab/api/?type=commit&cmd=%3Ccommit%3E%3Cpartial%3E$xpath%3C%2Fpa**
**rtial%3E%3C%2Fcommit%3E&key=$key**

# Job status changes to "FIN" when the job is done.
PS_C:\>**$pan=Invoke-RestMethod -Uri "https://pan-**
**lab/api/?type=op&cmd=%3Cshow%3E%3Cjobs%3E%3Cid%3E$jobid%3C%2Fi**
**d%3E%3C%2Fjobs%3E%3C%2Fshow%3E&key=$key"**
PS_C:\>**$pan.response.result.job.status**
FIN

# Verifies that changes have been applied.
# The settings were changed from alert to block.
PS_C:\>**$pan=Invoke-RestMethod -Uri "https://pan-**
**lab/api/?type=config&action=show&key=$key=&xpath=$xpath"**

Paul Mastad paulmastad@outlook.com

```
PS_C:\>$pan.response.result.entry
name                    action          wildfire-action
----                    ------          ---------------
http                    block           block

# Removed the key phrase from memory.
PS_C:\>$key="none"
```

## 4. Conclusion.

Smaller devices are already numerous, and their use – and, thus, proliferation – will continue. Like PCs and other larger devices, they too can face threats. Palo Alto Networks next-generation firewalls (PAN) can help in the fight against malicious Android apps. WildFire is a PAN feature that runs unknown files in a Virtual Environment. It observes behavior and makes it possible to detect zero-day malware or targeted malware. Palo Alto Networks version 6.0 has new features that uncover unknown malicious Android apps.

Palo Alto Networks next-generation firewalls are both powerful and complex. Using PAN to protect WiFi requires both careful consideration and the necessary skills. Organizations that already have adopted PAN will benefit from IT staff with existing PAN skills. Multiple smaller PANs can be used to protect selected WiFi home networks.

Scripting with the PAN API is a supplement or alternative to other administrative methods. PowerShell and PAN API are a means to administer single or multiple PANs. PowerShell commands function as both one-liners or as part of more complex scripts. Curl[2] and Wget[3] are alternatives to using PowerShell.

PAN, WildFire, and PAN API are building blocks. Risk analysis and careful planning are essential in order to turn the building blocks into a solution.

[2] http://curl.haxx.se/
[3] https://www.gnu.org/software/wget/

Paul Mastad paulmastad@outlook.com

References

Addicks, B. (2012, December 28). Scripting with Palo Alto Networks. *Lockstep Technology Group Blogs*. Retrieved from: http://blogs.lockstepgroup.com/2012/12/scripting-with-palo-alto-networks.html

Chou, J., Loverde, L., O'Donnell, B., & Shirer, M. (2013, May 28). PC Outlook falls as market increasingly looks to tablets, according to IDC. *IDC*. Retrieved from: http://www.idc.com/getdoc.jsp?containerId=prUS24129913

Combs, J. (2014, February 28). How to send multipart/form-data with PowerShell Invoke-RestMethod. Retrieved from: http://stackoverflow.com/questions/22491129/how-to-send-multipart-form-data-with-powershell-invoke-restmethod

Ettema,T. (2014, February 14). WildFire security overview. *Palo Alto Networks Live.* Retrieved from: https://live.paloaltonetworks.com/docs/DOC-2880

Forester (2014, June 16). Privacy and data protection by country. *Forester's Global Data Protection and Privacy Heatmap*. Retrieved from: https://www.forrestertools.com/heatmap/

F-Secure Labs (2014). *Threat report H2 2013.* F-Secure Labs.

GET (2014, June 09). *Hjem / TV / Tv på iPad og iPhone*. Retrieved from: http://translate.google.com/translate?depth=1&hl=en&ie=UTF8&prev=_t&rurl=translate.google.com&sl=auto&tl=en&u=http://www.get.no/produkter/tv/tv-p%25C3%25A5-ipad

Hill, K. (2013, April 22). PowerShell equivalent for cURL command uploading file. *Stack Overflow*. Retrieved from: http://stackoverflow.com/questions/16137665/powershell-equivalent-for-curl-command-uploading-file

Hoback, C., Khana, N. & J. Ramos (Producers) & Hoback, C. (Director). (2013). *Terms and conditions may apply* [Motion Picture]. United States: Hyrax Films.

Hyres, K. (2014, January 27). *Global smartphone shipments reach a record 990 million units in 2013*. *Strategy Analytics.* Retrieved from: http://blogs.strategyanalytics.com/WSS/post/2014/01/27/Global-Smartphone-Shipments-Reach-a-Record-990-Million-Units-in-2013.aspx

Paul Mastad paulmastad@outlook.com

Ilyin, Y. (2013, September 17). Mobile devices at work and home: the blurring borderline. *Kaspersky Business.* Retrieved from: http://business.kaspersky.com/mobile-devices-at-work-and-home-the-blurring-borderline/

Khalaf, S. (2014, April 22). The rise of the mobile addict. *Flurry blog.* Retrieved from: http://blog.flurry.com/bid/110166/The-Rise-of-the-Mobile-Addict

Luckerson, V. (2014, January 13). Tech's big promises. *Time Magazine Europe*, pp. 24-25.

Mainelli, T., Ubrani, J., Reith, R., & Shirer, M. (2014, January 29). A strong holiday quarter for the worldwide tablet market, but signs of slower growth are clear, according to IDC. *IDC*. Retrieved from: http://www.idc.com/getdoc.jsp?containerId=prUS24650614

Messmer, E. (2012, November 13). Palo Alto Networks targets VMware shops with virtualized next-gen firewalls. *Network World.* Retrieved from: http://www.networkworld.com/news/2012/111312-palo-alto-virtual-264196.html

Microsoft Technet (2013, October 17). Invoke-RestMethod. *Technet.* Retrieved from: http://technet.microsoft.com/en-us/Library/hh849971.aspx

Microsoft Technet (2013, October 17). Scripting with Windows PowerShell. *Technet.* Retrieved from: http://technet.microsoft.com/en-us/library/bb978526.aspx

Microsoft Windows Server Team (2009, July 22). Windows Server 2008 R2 reaches the RTM milestone! *Technet.* Retrieved from: http://blogs.technet.com/b/windowsserver/archive/2009/07/22/windows-server-2008-r2-rtm.aspx

Palo Alto Networks (2011, May 18). Palo Alto Networks in the data center. *Palo Alto Networks.* Retrieved from: https://www.paloaltonetworks.com/resources/whitepapers/palo-alto-networks-in-the-data-center.html

Palo Alto Networks (2014, May 26). Network security solutions by Palo Alto Networks. *Palo Alto Networks.* Retrieved from: https://www.paloaltonetworks.com/solutions.html

Paul Mastad paulmastad@outlook.com

Palo Alto Networks (2014, February 20). PAN OS and Panorama XML API Reference Guide 6.0. *Palo Alto Networks.* Retrieved from: https://live.paloaltonetworks.com/docs/DOC-6607

Palo Alto Networks (2014, February 26). WildFire API programming guide. *WildFire Portal*. Retrieved from: https://wildfire.paloaltonetworks.com/wildfire/guide (requires login)

Palo Alto Networks (2014, June 11). WildFire: Dynamic analysis to identify and block unknown threats. *Palo Alto Networks.* Retrieved from: https://www.paloaltonetworks.com/content/dam/paloaltonetworks-com/en_US/assets/pdf/datasheets/wildfire/wildfire.pdf

Palo Alto Networks Technical Publications (2012, November 1). PAN-OS-5.0.0-RN-revA.pdf. *Palo Alto Networks.* Retrieved from: https://downloads.paloaltonetworks.com/software/PAN-OS-5.0.0-RN-revA.pdf?__gda__=1398569416_93c3889446fa8dd9d74160f67ee5ddb2

Palo Alto Networks Technical Publications (2014, March 8). PAN-OS administrator's guide 6.0 (English). *Palo Alto Networks.* Retrieved from: https://live.paloaltonetworks.com/docs/DOC-6603

Palo Alto Networks Technical Publications (2014, April 29). PAN-OS Release Notes Version 5.0.12. *Palo Alto Networks.* Retrieved from: https://downloads.paloaltonetworks.com/software/PAN-OS-5.0.12-RN.pdf?__gda__=1401164115_2910014dc1047c96f0e243010de72fb8

Palo Alto Networks Technical Publications (2014, March 14). WildFire administrator's guide 6.0 (English). *Palo Alto Networks.* Retrieved from live.paloaltonetworks.com: https://live.paloaltonetworks.com/docs/DOC-6589

peli0. (2013, January 13). Sensor simulator. *OpenIntents*. Retrieved from: http://code.google.com/p/openintents/wiki/SensorSimulator

Petsas, T., Voyatzis, G., Athanasopoulos, E., Polychronakis, M., & Ioannidis, S. (2014, April 24). *Rage against the virtual machine: hindering dynamic analysis of android malware.* Proceedings from EuroSec '14: *Proceedings of the Seventh European Workshop on System Security*. New York, NY. Retrieved from: syssec-

project.eu: http://www.syssec-project.eu/m/page-media/3/petsas_rage_%20against_the_virtual_machine.pdf

Pietrek, M. (1994, March). Peering inside the PE: A tour of the Win32 portable executable file format. Retrieved from msdn: http://msdn.microsoft.com/en-us/library/ms809762.aspx

Polo, S. (2010, April 7). How to configure virtual wire. *Palo Alto Networks.* Retrieved from live.paloaltonetworks.com: https://live.paloaltonetworks.com/videos/1005

Rutkowska, J. (2006, June 22). Introducing blue pill. *The Invisible Things Lab's blog.* Retrieved from: http://theinvisiblethings.blogspot.no/2006/06/introducing-blue-pill.html

Symantec Corporation. (2014). Internet security threat report 2014: Volume 19. *Symantec.* Retrieved from: http://www.symantec.com/security_response/publications/threatreport.jsp

ukhapre (2014, February 7). How to enable CLI debug. *Palo Alto Networks.* Retrieved from: https://live.paloaltonetworks.com/docs/DOC-5995

virustotal. (2014, March 3). *SHA-256: 2ce2ee9fdcc263bd5e9afe79d1cc93025b5c8d0ef8c9c3fc5912cd08cfe0c743*. Retrieved from virustotal: https://www.virustotal.com/en/file/2ce2ee9fdcc263bd5e9afe79d1cc93025b5c8d0ef8c9c3fc5912cd08cfe0c743/analysis/

Wright, E. (2014, February 26). PowerShell – Invoke-RestMethod: Putting the cURL in your shell. *DiscoPosse.* Retrieved from: http://www.discoposse.com/index.php/2012/06/30/powershell-invoke-restmethod-putting-the-curl-in-your-shell/

Xu, Z., Ciao, C., & Olson, R. (2014, May 12). Funtasy trojan targets Spanish Android users with sneaky SMS charges. *Palo Alto Networks.* Retrieved from: http://researchcenter.paloaltonetworks.com/2014/05/funtasy-trojan-targets-spanish-android-users-sneaky-sms-charges/#more-5544

Paul Mastad paulmastad@outlook.com

# 5. Appendix A - Specimen 1 - Report via E-mail.

From: report@wildfire.paloaltonetworks.com [mailto:report@wildfire.paloaltonetworks.com]
Sent: xx April 2014 xx:xx1
To:
Subject: Your WildFire analysis report is ready

WildFire Analysis Report

File Name: TT Copy1.scr
Uploaded by: pan-lab (S/N xxx) at 2014-04-xx xx:xx:xx BST
SHA256: xxx
File URL: unknown
User: unknown
Application: xx
Source IP/Port: xxx.xxx.xxx.xxx:49253
Destination IP/Port: xxx.xxx.xxx.xxx:xx

Verdict: This sample was determined to be malware.
Summary of behaviors observed during analysis:
- Created or modified files
- Spawned new processes
- Contained unknown TCP/UDP traffic
- Modified Windows registries
- Modified registries or system configuration to enable auto start capablity
- Changed security settings of Internet Explorer
- Visited a known dynamic DNS domain
- Created a file in the Windows folder
- Created an executable file in a user document folder
- Started a process from a user document folder
- Changed the Windows firewall policy
- Installed a service
- Registered a file as auto-start from a local directory
- Attempted to sleep for a long period
- Sample attempted to copy itself

The detailed forensics report can be viewed at:
https://wildfire.paloaltonetworks.com/report/box/xxx/xxx

# 6. Appendix B - Specimen 2 – WildFire Analysis Report

## 1. File Information

Paul Mastad paulmastad@outlook.com

| | |
|---|---|
| File Type | Android APK |
| File Signer | nn |
| SHA-256 | xxxx |
| MD5 | xxxx |
| File Size | 1000000 bytes |
| First Seen Timestamp | 2014-02-16 05:47:59 PST |
| Verdict | Benign |
| Antivirus Coverage | VirusTotal Information |

| | |
|---|---|
| File Type | APK |
| APK Package Name | no.xx.app |
| APK Version | 1.2 |
| Min SDK Requirement | 8 |
| Max SDK Requirement | |
| Target SDK | 17 |

# 2. Static Analysis

## 2.1. Package Information

## 2.2. Suspicious File Properties

This sample was not found to contain any high-risk content during a pre-screening analysis of the sample.

| Behavior |
|---|
| The APK file contains URLs within the source code. |

## 2.2. Static Attributes

| Defined Activities |
|---|
| no.xxxx.app.XxxxMain |
| notification.NotificationActivity |
| Defined Services |
| .GCMIntentService |
| Defined Intent Filters |
| android.intent.action.MAIN |
| android.intent.action.VIEW |
| com.google.android.c2dm.intent.RECEIVE |
| com.google.android.c2dm.intent.REGISTRATION |
| Defined Receivers |
| com.google.android.gcm.GCMBroadcastReceiver |
| Requested Permissions |
| no.xxxx.app.permission.C2D_MESSAGE |

Paul Mastad paulmastad@outlook.com

| |
|---|
| com.google.android.c2dm.permission.RECEIVE |
| android.permission.ACCESS_WIFI_STATE |
| android.permission.ACCESS_COARSE_LOCATION |
| android.permission.INTERNET |
| android.permission.ACCESS_NETWORK_STATE |
| android.permission.CHANGE_NETWORK_STATE |
| android.permission.GET_ACCOUNTS |
| android.permission.WAKE_LOCK |
| android.permission.READ_PHONE_STATE |
| Sensitive API Calls Performed |
| android/net/ConnectivityManager;->getNetworkInfo |
| android/telephony/TelephonyManager;->getDeviceId |
| android/accounts/AccountManager;->getAccounts |
| android/net/wifi/WifiManager;->getConnectionInfo |
| android/net/ConnectivityManager;->getActiveNetworkInfo |
| android/webkit/GeolocationPermissions$Callback;->invok |
| android/app/NotificationManager;->notify |
| Defined Sensors |
| Receive sensor readings from gps |

## 2.4. Embedded URLs

| URL | File Location | Known Malicious URL |
|---|---|---|
| http://www.apache.org/licenses/LICENSE-2.0 | /tmp/537696025/smali/assets/fonts/roboto.ttf | |
| http://www.fontfont.com | /tmp/537696025/smali/assets/fonts/clanot.otf | |
| http://www.fontfont.com/eula/license.html | /tmp/537696025/smali/assets/fonts/clanot.otf | |
| http://crl.verisign.com/tss-ca.crl0 | /tmp/537696025/smali/assets/fonts/clanot.otf | |
| http://ocsp.verisign.com0 | /tmp/537696025/smali/assets/fonts/clanot.otf | |
| http://crl.verisign.com/ThawteTimestampingCA.crl0 | /tmp/537696025/smali/assets/fonts/clanot.otf | |

## 2.5. Suspicious API Calls

| API Calls | File Location | Description |
|---|---|---|
| android/net/ConnectivityManager;->getActiveNetworkInfo | /tmp/537696025/smali/smali/collection/Internet.smali | Sensitive API call that should not normally be called by typical apps |
| android/net/ConnectivityManager;->getActiveNetworkInfo | /tmp/537696025/smali/smali/android/support/v4/net/ConnectivityManagerCompatGingerbread.smali | Sensitive API call that should not normally be called by typical apps |
| android/net/ConnectivityManager;->getActiveNetworkInfo | /tmp/537696025/smali/smali/android/support/v4/net/ConnectivityManagerCompat$BaseConnectivityManagerCompatImpl.smali | Sensitive API call that should not normally be called by typical apps |
| android/net/ConnectivityManager;->getNetworkInfo | /tmp/537696025/smali/smali/android/support/v4/net/ConnectivityManagerCompat.smali | Sensitive API call that should not normally be called by typical apps |
| android/net/ConnectivityManager;->getActiveNetworkInfo | /tmp/537696025/smali/smali/android/support/v4/net/ConnectivityManagerCompatHoneycombMR2.smali | Sensitive API call that should not normally be called by typical apps |

Paul Mastad paulmastad@outlook.com

| | | |
|---|---|---|
| android/app/NotificationManager;->notify | /tmp/537696025/smali/smali/no/xxxx/app/GCMIntentService.smali | Sensitive API call that should not normally be called by typical apps |

# 3. Dynamic Analysis

## 3.1. VM1 (Android 2.3, API 10, avd2.3.1)

### 3.1.1. Behavioral Summary

This sample was found to be **benign** on this virtual machine.

| Behavior |
|---|
| Sample used SSL |
| The APK file attempts to connect to a URL. |
| Sample used a New User-Agent |
| Performed a failed HTTP connection |

### 3.1.2. Network Activity

#### DNS Queries

| Domain Name | Query Type | DNS Response |
|---|---|---|
| mmetrix.mobi | NS | ns1.crystone.net |
| g.akamaiedge.net | NS | n6g.akamaiedge.net |
| mmetrix.mobi | NS | ns2.crystone.net |
| googleusercontent.com | NS | ns2.google.com |
| googleusercontent.com | NS | ns3.google.com |
| payment.xxxx.no | A | 195.88.xxx.xxx |
| googleusercontent.com | NS | ns1.google.com |
| g.akamaiedge.net | NS | n2g.akamaiedge.net |
| xx.mnocdn.no | A | 80.91.xxx.xxx |
| g.akamaiedge.net | NS | n3g.akamaiedge.net |
| mnocdn.no | NS | dns1.xxxx-it.no |
| mnocdn.no | NS | dns2.xxxx-it.no |
| ssl.google-analytics.com | A | 173.194.40.158 |
| g.akamaiedge.net | NS | n1g.akamaiedge.net |
| g.akamaiedge.net | NS | n4g.akamaiedge.net |
| googleusercontent.com | NS | ns4.google.com |

#### HTTP Requests

| HTTP Method | URL | User-Agent |
|---|---|---|
| GET | m.xxxx.no/?service=cssMobile&publication=xxxx&v=30&r=9a44b67682e073cdb79e3a12b7e98ad69f393839 | Mozilla/5.0 (Linux; Android 4.4; Nexus 5 Build/BuildID) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/30.0.0.0 Mobile Safari/537.36 AndroidWrapper |
| GET | ap.mnocdn.no/incoming/article7468440.ece/ALTERNATES/w380c169/IMG_8780.jpg?updated=130220141048.jpg | Mozilla/5.0 (Linux; Android 4.4; Nexus 5 Build/BuildID) AppleWebKit/537.36 (KHTML, like |

| | | |
|---|---|---|
| | | Gecko) Version/4.0 Chrome/30.0.0.0 Mobile Safari/537.36 AndroidWrapper |
| GET | m.xxxx.no/?service=cssMobile&widgets=widgets/code/html;widgets/code/jsp;widgets/eventPlaceSearch/default;widgets/eventSearch/advancedSearch;widgets/list/simple;widgets/mobileAd/default;widgets/mobileStories/default;widgets/offCanvas/offCanvasDefault;widgets/rating/ratingList;widgets/statistics/xiti;widgets/stories/ledBanner;widgets/switchMaster/default;&v=30&publication=xxxx&r=9a44b67682e073cdb79e3a12b7e98ad69f393839 | Mozilla/5.0 (Linux; Android 4.4; Nexus 5 Build/BuildID) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/30.0.0.0 Mobile Safari/537.36 AndroidWrapper |
| GET | m.xxxx.no/?hideTopBottom=true | Mozilla/5.0 (Linux; Android 4.4; Nexus 5 Build/BuildID) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/30.0.0.0 Mobile Safari/537.36 AndroidWrapper |
| GET | ap.mnocdn.no/incoming/article7472279.ece/ALTERNATES/w180c169/000671363-frYmENNaWn.jpg?updated=160220141318.jpg | Mozilla/5.0 (Linux; Android 4.4; Nexus 5 Build/BuildID) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/30.0.0.0 Mobile Safari/537.36 AndroidWrapper |
| GET | ap.mnocdn.no/incoming/article7471654.ece/ALTERNATES/w180c169/afp000629446.jpg?updated=150220141232.jpg | Mozilla/5.0 (Linux; Android 4.4; Nexus 5 Build/BuildID) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/30.0.0.0 Mobile Safari/537.36 AndroidWrapper |
| GET | cdn.mxpnl.com/libs/mixpanel-2.2.min.js | Mozilla/5.0 (Linux; Android 4.4; Nexus 5 Build/BuildID) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/30.0.0.0 Mobile Safari/537.36 AndroidWrapper |
| GET | ap.mnocdn.no/incoming/article7470428.ece/ALTERNATES/w380c169/afp000668160.jpg?updated=140220141258.jpg | Mozilla/5.0 (Linux; Android 4.4; Nexus 5 Build/BuildID) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/30.0.0.0 Mobile Safari/537.36 AndroidWrapper |
| GET | mm.xxxx.no/projects/xxxx/appmenu/json.php?os=android | Apache-HttpClient/UNAVAILABLE (java 1.4) |
| GET | m.xxxx.no/?service=cssMobile&widgets=widgets/code/html;widgets/code/jsp;widgets/eventPlaceSearch/default;widgets/eventSearch/advancedSearch;widgets/list/simple;widgets/mobileAd/default;widgets/mobileStories/default;widgets/offCanvas/offCanvasDefault;widgets/rating/ratingList;widgets/statistics/xiti;widgets/stories/ledBanner;widgets/switchMaster/default;&v=30&publication=xxxx&r=9a44b67682e073cdb79e3a12b7e98ad69f393839 | Mozilla/5.0 (Linux; Android 4.4; Nexus 5 Build/BuildID) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/30.0.0.0 Mobile Safari/537.36 AndroidWrapper |
| GET | mmetrix.mobi/tns_msr.js | Mozilla/5.0 (Linux; Android 4.4; Nexus 5 Build/BuildID) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/30.0.0.0 Mobile Safari/537.36 AndroidWrapper |
| GET | www.xxxx.no/resources/js/mno/xiti//xtclicks.js | Mozilla/5.0 (Linux; Android 4.4; Nexus 5 Build/BuildID) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/30.0.0.0 Mobile Safari/537.36 AndroidWrapper |
| GET | m.xxxx.no/?hideTopBottom=true | Mozilla/5.0 (Linux; Android 4.4; Nexus 5 Build/BuildID) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/30.0.0.0 Mobile Safari/537.36 AndroidWrapper |
| GET | ap.mnocdn.no/incoming/article7472279.ece/ALTERNATES/w180c169/000671363-frYmENNaWn.jpg?updated=160220141318.jpg | Mozilla/5.0 (Linux; Android 4.4; Nexus 5 Build/BuildID) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/30.0.0.0 Mobile Safari/537.36 AndroidWrapper |
| GET | ap.mnocdn.no/incoming/article7471654.ece/ALTERNATES/w180c169/afp000629446.jpg?updated=150220141232.jpg | Mozilla/5.0 (Linux; Android 4.4; Nexus 5 Build/BuildID) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/30.0.0.0 Mobile Safari/537.36 AndroidWrapper |

Paul Mastad paulmastad@outlook.com

| | | |
|---|---|---|
| GET | cdn.mxpnl.com/libs/mixpanel-2.2.min.js | Mozilla/5.0 (Linux; Android 4.4; Nexus 5 Build/BuildID) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/30.0.0.0 Mobile Safari/537.36 AndroidWrapper |
| GET | ap.mnocdn.no/incoming/article7470428.ece/ALTERNATES/w380c169 /afp000668160.jpg?updated=140220141258.jpg | Mozilla/5.0 (Linux; Android 4.4; Nexus 5 Build/BuildID) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/30.0.0.0 Mobile Safari/537.36 AndroidWrapper |

**Connections**

| Host | Port | Protocol | Country |
|---|---|---|---|
| xxx.xxx.xxx.xxx | 80 | TCP | NO |
| xxx.xxx.xxx.xxx | 80 | TCP | SE |
| xxx.xxx.xxx.xxx | 443 | TCP | NO |
| xxx.xxx.xxx.xxx | 80 | TCP | N/A |

# 7. Appendix C - Template.

D:\source\input.txt
A template to use in queries against WildFire-in-the-cloud.

----------0a8d842719f1707d
Content-Disposition: form-data; name="SHA-256"

ReplaceThisStringWithSHA-256
----------0a8d842719f1707d
Content-Disposition: form-data; name="apikey"

ReplaceThisStringWithApikey
----------0a8d842719f1707d--

Paul Mastad paulmastad@outlook.com