ÖN BİLGİ

Bu belge

- https://www.bgasecurity.com/makale/kurumsal-aglarda-log-inceleme-yontemiyle-saldiri-analizi/

resmi adresindeki veya linkin kırık olması ihtimaline karşın alternatif olarak

- https://www.includekarabuk.com/kitaplik/indirmeDeposu/Siber_Guvenlik_Teknik_Makaleler/Teori/BaskalarinaAitMakaleler/Kurumsal%20A%C4%9Flarda%20Log%20%C4%B0nceleme%20Y%C3%B6ntemiyle%20Sald%C4%B1r%C4%B1%20Analizi.pdf

adresindeki makaleye çalışılarak elde edilen notlarımı kapsamaktadır. Bu çıkarılan notlar belgemde alıntılar ve/veya kişisel ilavelerim mevcuttur.

1)

APT Saldırısı Nedir?
An advanced persistent threat (APT) is a network attack in which an unauthorized person gains access to a network and stays there undetected for a long period of time. The intention of an APT attack is to steal data rather than to cause damage to the network or organization. APT attacks target organizations in sectors with high-value information, such as national defense, manufacturing and the financial industry.

(http://searchsecurity.techtarget.com/definition/advanced-persistent-threat-APT)

2)

Bir kişinin ya da kurumun son bir yıl içerisinde APT benzeri sofistike bir saldırıya maruz kalıp kalmadığını anlamanın yolu log alt yapısının varlığı ve bu log'ları analiz edebilme kabiliyetidir.

(Page 4)

3)

Log yönetimi ile log toplama ayrı şeylerdir. Log yönetiminin olabilmesi için öncelikle sağlıklı bir log toplama mekanizmasının kurulu olması gerekmektedir. Toplanan log'lardan anlamlı sonuçlar üretecek işlemler gerçekleşiyorsa log toplamadan log yönetimine doğru geçiş yapılıyor demektir.

(page 6)

4)

Aynı anda onlarca porta yönelik gelen SYN paketlerinin herbirini ayrı ayrı log'lamak yerine tek bir satırda gösterip Port Tarama olarak yazmak bir Log Korelasyonu örneğidir.

(page 7)

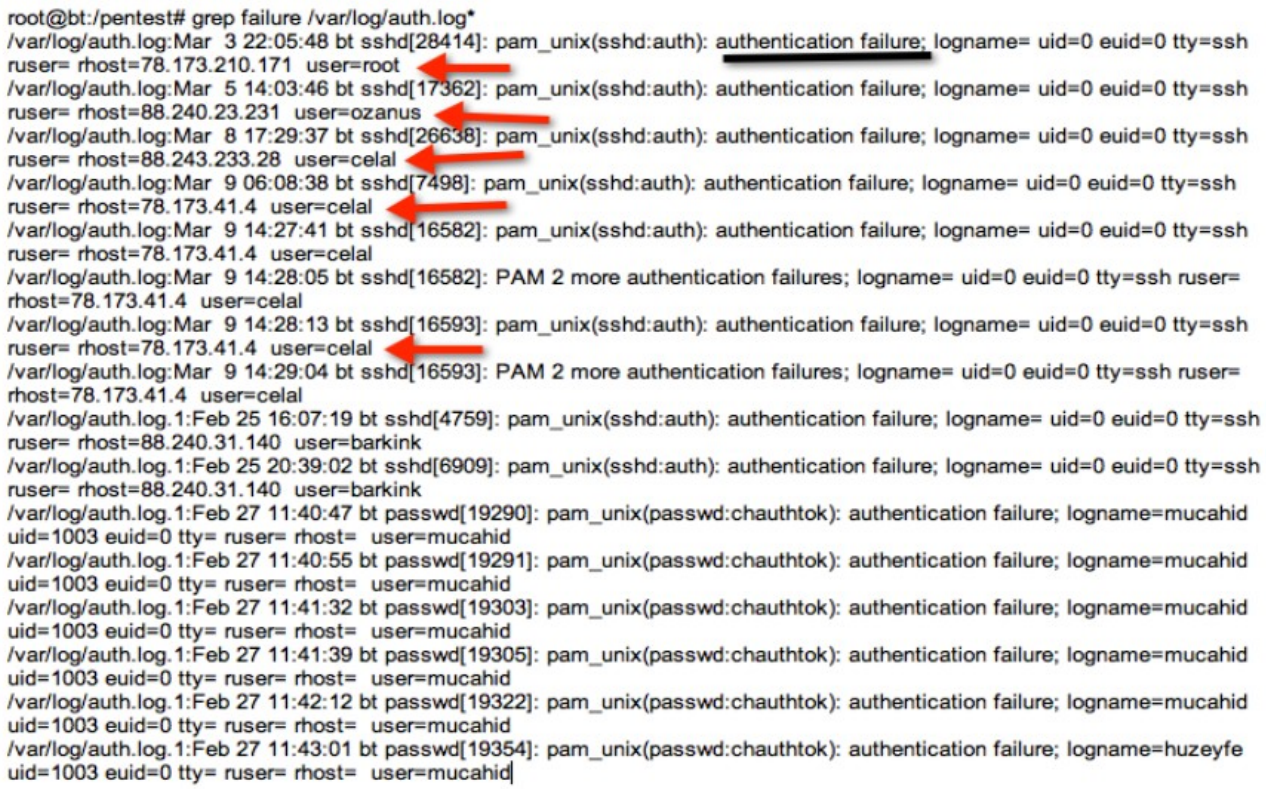
5)

Genellikle UNIX sistem yöneticileri her yerden sistemlerine erişebilmek için SSH servisini dışarı açık bırakırlar. Nmap tabiriyle Open olan bu SSH servisleri hacker'ların ilgisini çeker ve SSH parolası için Brute Force denemeleri yaparlar.

(Page 8)

6)

Aşağıda SSH Brute Force saldırısına maruz kalan sistemin log kayıtlarını görmektesiniz.



Görüldüğü üzere sürekli bir login denemesi yapılmış ve hep Authentication Failure yanıtı dönmüş.

(Page 9)

7)

Birçok kurumda log'lar sadece kayıt altına alınmaktadır. Fakat log analizi yapılmamaktadır. Log'lama tek başına bir değer ifade etmez. Nasıl ki tarladan toplanan hasat işlenmeden bir işe yaramazsa koleksiyon amacı ile toplanan ve ne izlenmeye ve ne de değerlendirmeye tutulmayan log'lar da bir şey ifade etmez.

(Page 12)

8)

Log demek iz kaydı demektir.

(page 12)

9)

Log analizi teknik bir işin ötesinde ilgili konuya dair ciddi bir tecrübe ve düşünme yetisi ister. Hangi saldırı tipi gerçekleştirilmiş sorusunun cevabını bulabilmek için saldırı kavramını ve detaylarını iyi bilmek gerekir.

(Page 12)

10)

Farklı amaçlar için kullanılabilecek onlarca ticari ve açık kaynak kodlu ücretsiz log analiz yazılımları bulunmaktadır. Log analiz yazılımları genellikle arka planda bir veritabanı uygulaması kullanarak işlem yaparlar. Log analiz yazılımlarını doğru kullanabilmek için log'lar içerisinde neyi aradığımızı bilmek önemlidir. Örneğin bir saldırı arıyorsak web tabanlı bir saldırı mı yoksa network tabanlı bir saldırı mı arıyoruz gibi daha sepesifik bir hedef tanımlamamız gerekir.

(Page 13)

11)

Linux işletim sistemi dosya temelli olması nedeniyle dosya işleme amaçlı onlarca araç barındırmaktadır. Log analizinde de kullanılabilen linux araçları şunlardır:

● cut
- Prints selected parts of lines from a FILE to standard output.

Example:

records
Johnson Sara 21
Smith Tom 32
Jones Mindy 25
Anderson Bob 42

```
> cut -d ' ' -f 2 records > lastnames          // d means delimiter, f means field
> cat lastnames
```

Output:
Sara
Tom
Mindy
Bob

- awk

  - Scan pattern and does text processing. It is more advanced than cut.

  Examples

    records

      Johnson Sara 21
      Smith Tom 32
      Jones Mindy 25
      Anderson Bob 42

    > cat records | awk ' {print $1,$3} ' // Birinci ve üçüncü kolon

    Output:

      Johnson 21
      Smith 32
      Jones 25
      Anderson 42

    > cat records | awk ' /^Jo/ ' // Jo ile başlayanlar | Syntax : ' /pattern/ '

    Output:

      Johnson Sara 21
      Jones Mindy 25

    > cat records | awk ' /^Jo/ {print $1} ' // Jo ile başlayanların ilk kolonu

    Output:

      Johnson
      Jones

- grep

  - Print lines which matches with specified pattern.

  Example

    file.txt

      THIS LINE IS THE 1ST UPPER CASE LINE IN THIS FILE.
      this line is the 1st lower case line in this file.
      This Line Has All Its First Character Of The Word With Upper Case.

      Two lines above this line is empty.
      And this is the last line.

    > grep --color -n "this" file.txt // -n: line number

    Output:

      2:this line is the 1st lower case line in this file.
      5:Two lines above this line is empty.
      6:And this is the last line.

● more
- Displays text, one screen at a time **only** through forward.

● less
- Displays text, one screen at a time both through forward and backward with arrow keys. Also the program less does not require the whole file to be loaded in memory to view parts of it. Therefore it starts up faster on large files than editors.

● tail
- Outputs the last parts of file.

Examples

```
// Son 10 satırı yazdırır.
tail wordlist.txt

// Son 100 satırı yazdırır.
tail -n 100 wordlist.txt

// Son 100 satırı yazdırır, fakat log.txt'e eklenen her yeni
// kaydı ekranda anlık olarak gösterir. Log takibinde
// sniffer'dan gelen paketlerin ve credentials'ların takibinde idealdir.
> tail -f -n 100 log.txt
```

● head
- Outputs the first parts of file.

Example

```
// İlk 100 satırı yazdırır.
> head -n 100 wordlist.txt
```

● cat
- Outputs the content of the file

● sort
- Sorts lines of file

Examples

```
data.txt
        pears
        apples
        oranges

> sort data.txt          // Alfabetik sıralar

Output:
        apples
        oranges
        pears
```

● uniq

- Filters duplicate lines

Examples

data.txt

```
This is a line.
This is a line.
This is a line.

This is also a line.
This is also a line.

This is also also a line.
```

> uniq data.txt

Output:

```
This is a line.

This is also a line.

This is also also a line.
```

> uniq -c data.txt

Output:

```
3 This is a line.              // 3 tane bu satırdan varmış.
1                              // 1 tane boş satır varmış.
2 This is also a line.         // 2 tane bu satırdan varmış.
1                              // 1 tane boş satır varmış.
1 This is also also a line.    // 1 tane bu satırdan varmış.
```

● strings

- Prints strings in **binary** and **nonascii** files.

Example

```
> gcc main.c -o main
> strings main | grep return        // It searches “return” keyword in main object
```

(http://serverfault.com/questions/51477/linux-command-to-find-strings-in-binary-or-non-ascii-file)

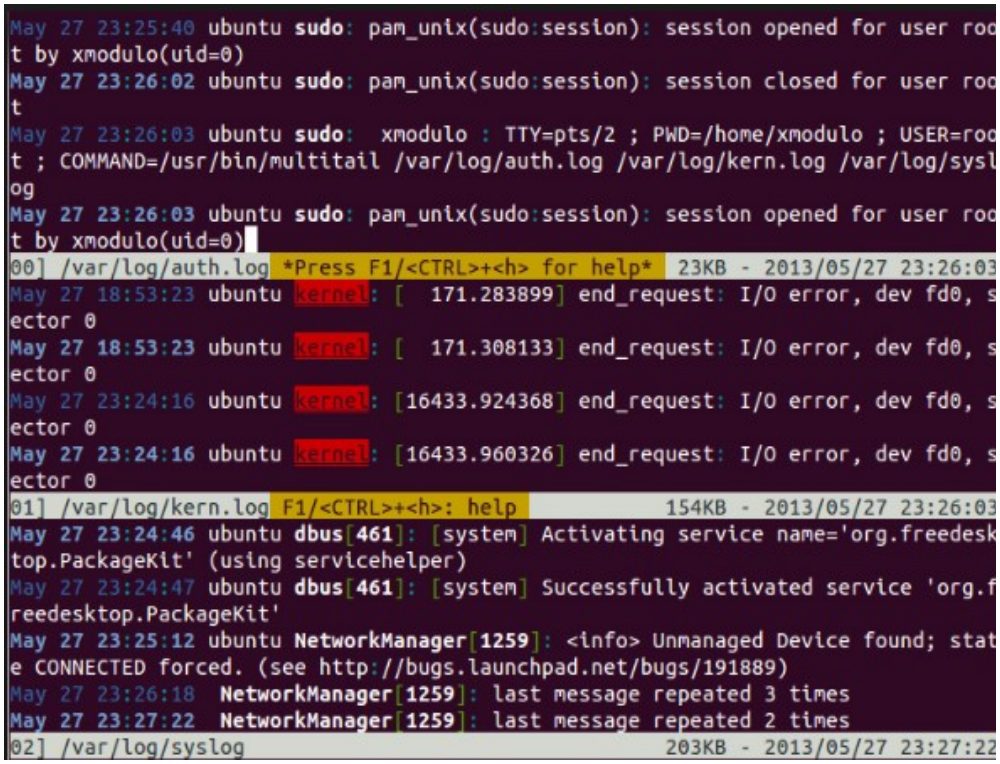
● multitail

- Outputs the last parts of multiple file synchronizely.

Example

> multitail -f /var/log/syslog /var/log/auth.log /var/log/kern.log

Output:



Or we can use tail tool for multiple files as well:

> tail -f /var/log/syslog -f /var/log/auth.log -f /var/log/apache.log

(Page 15)

12)

Log analizinde kullanılan linux araçları hızlıdır, esnektir ve performanslıdır. Fakat veritabanı mantığı olmadığı için her işlemde tekrar aynı süreç yaşanır. Ayrıca çok büyük log dosyalarının (big data'nın) incelenmesinde zaman ve performans kaybına yol açar.

(page 15)

13)

Log Analiz Metodolojisi
Log analizinde kullanılacak metodoloji şu şekildedir:

I. Adım
- Log dosyalarının elde edilmesi
- Log dosyalarının normalleştirilmesi

II. Adım
- Log analiz amacının belirlenmesi
- Log analiz amacının teknik dile çevrimi

III. Adım
- Log analiz yazılımı kullanarak istenen verileri ayıklama
- İstenen sonuçların elde edilip edilemediğine dair doğrulama

(Page 17)

14)

Encoding

• Bir veriyi başka formatlarda gösterme işlemine denir.
• Geriye çevrilebilir algoritmalardır.
• Base64, URL Encoding, Hex Encoding en sık karşılaşılan encoding algoritmalarındandır.

(Page 19)

15)

Apache ve diğer web sunucu yazılımları öntanımlı olarak POST isteklerinden gelen değerleri log'lamazlar. Bunun için mod-forensic gibi ya da mod_security gibi ek bileşenler kullanılmalıdır. Ayrıca web sunuculara yönelik POST üzerinden gerçekleşebilecek saldırıları yakalamak için WAF, Load Balancer, IPS gibi ürünlerin log'larına da başvurmak gerekebilir.

(Page 23)

16)

Diyelim ki DVWA'nın login panelinden login olacağız. Bu durumda network'te akan trafiği dinleseydik login panelindeki butonu tıkladığımızda şöyle bir paket içeriğiyle karşılaşacaktık:

T 127.0.0.1:47635 -> 127.0.0.1:80 [AP]
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:5.0.1) Gecko/20100101 Firefox/5.0.1
Accept: text/html, application/xhtml+xml, application/xml; q=0.9,*/*;q=0.8
Accept-Language: en-us, en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1, utf-8; q=0.7,*;q=0.7
Connection: keep-alive
Referrer: http:/localhost/dvwa/login.php
Cookie: security=high; PHPSESSID=i68o2ejvm6jnp3b9pv083i4mi7;
Content-Type: application/x-www-form-urlencoded
Content-Length: 49

username=admin&password=sifre&Login=Login

Görüldüğü üzere POST edilen değişkenleri gözlemleyebildik. Fakat DVWA'yı kurduğumuz sistemde log kayıt tutma hizmeti aktifken log kayıtlarına bakacak olsaydık o paket log'lara şöyle düşecekti:

127.0.0.1 [04/Mar/2012:02:10:10-05:00] POST /dvwa/login.php HTTP/1.1 302 “http://localhost/dvwa/login.php” Mozilla/5.0 (X11; Linux i686; rv:5.0.1) Gecko/20100101 Firefox/5.0.1

Yani yukarıdakini tek satır (record olarak) düşünecek olursak açtığımızda şöyle olacaktır:

127.0.0.1
[04/Mar/2012:02:10:10-05:00]
POST /dvwa/login.php HTTP/1.1
302 “http://localhost/dvwa/login.php” Mozilla/5.0 (X11; Linux i686; rv:5.0.1) Gecko/20100101 Firefox/5.0.1

Görüldüğü üzere log'a yansıyan kayıtta POST edilen değişkenler yer almamıştır. Bunun tek nedeni web sunucusunun standart yapılandırmada bırakılmış olmasındandır.

(Page 24-25)

17)

Sql Injection

• Uygulamaların veritabanı ile ilişkili olan kısımlarının manipule edilerek veritabanında sorgu çalıştırma işlemine SQLi denir.

• Veritabanıyla ilişkili olan tüm girdi noktalarında SQLi zafiyeti bulunabilir.

• SQL Injection saldırıları ile hedef sistemdeki kimlik doğrulama (login) adımı aşılabilir, hedef sistemdeki veritabanına ait tablolar ve hedef sistemde yüklü dosyalardaki bilgilere erişilebilir ve hatta hedef sistem ele geçirilebilir.

• SQL Injection saldırıları web üzerinden yapılsa da aslında saldırının istismar aşaması tamamen veritabanına yöneliktir.

• En fazla görülen ve en tehlikeli saldırı tiplerinden biridir.

• HTTP GET kullanılarak yapılan Sql Injection saldırıları log'larda olduğu gibi gözükebilecektir.

• Bazı durumlarda saldırgan çeşitli encoding teknikleri kullanarak saldırıyı gizlemeye çalışır (Benim NOT: Kafamda somut örneği yok bu cümlenin.)

(Page 26)

18)

Daha önceden de belirtildiği gibi POST üzerinden giden paketlerdeki POST edilen değişkenler web sunucu log'larına düşmüyordu. Dolayısıyla POST üzerinden gerçekleştirilen SQL Injection saldırılarındaki payload'lar da (sql injection kodları da) log'larda görünmeyecektir.

(Page 31)

19)

URL encode edilmiş bir SQL Injection saldırısına ait paket örneği:

```
T 127.0.0.1:47635 -> 127.0.0.1:80 [AP]
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:5.0.1) Gecko/20100101 Firefox/5.0.1
Accept: text/html, application/xhtml+xml, application/xml; q=0.9,*/*;q=0.8
Accept-Language: en-us, en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1, utf-8; q=0.7,*;q=0.7
Connection: keep-alive
Referrer: http:/localhost/dvwa/login.php
Cookie: security=high; PHPSESSID=i68o2ejvm6jnp3b9pv083i4mi7;
Content-Type: application/x-www-form-urlencoded
Content-Length: 49
```

username=ECYH%252C%2528SELECT%2520%2528CASE%2520WHEN%2520%25282167%253D2167%2529%2520THEN%2520ECYH%2520ELSE%25202167%252A%2528SELECT%25202167%2520FROM%2520INFORMATION_SCHEMA.CHARACTER_SETS%2529%2520END%2529&password=parola&Login=Login

Yukarıda POST'lanan veri sunucuda decode edildiğinde aşağıdaki şablonda bir SQL sorgusu ortaya çıkacaktır.

ECHO SELECT …. CASE ….. WHEN …. THEN ECYH ELSE SELECT FROM INFORMATION_SCHEMA.CHARACTER_SETS AND

Bu tip POST isteği kullanılarak gerçekleştirilen saldırılar Ağ Tabanlı IPS/IDS ya da WAF/Load Balancer sistemleri kullanarak tespit edilebilmektedir.

(Page 30-32)

19)

Sadece GET üzerinden denenen SQL Injection saldırılarında log'a düşen kayıtları saptamak için tercih edilen yöntem sql injeciton için kullanılan keyword'lerin ve özel karakterlerin log kayıtlarında aratılmasıdır. Örneğin concat, char, union, select, order by, group by gibi komutlar log kayıtları arasında aratılarak bir saldırı var mı yok mu tespit edilebilir. Bu arada bu kelimeleri log dosyasında aratmak false positive (hatalı pozitif) sonuçlar çıkarabileceği için çıkan sonuçların teker teker incelenmesi gerekir. Çünkü bu keyword'ler normal bir sql sorgusunda da kullanılmış olabilir.

(Page 33)

20)

SQL Injection (SQLi) Belirleme

> log.txt | grep -E 'concat|union' // concat ve union içeren satırları ekrana basar.

(Page 34)

21)

Scalp Tool'u
Scalp web sunucu log'larındaki saldırı izlerini tespit etmeye yarayan bir araçtır. Saldırı izlerini tespit etme analizinde kullanacağı değişkenleri de PHPIDS'den alır.

Benim NOT: Scalp.py tool'u Kali'de yüklü değil.

(Page 35)

22)

What is PHPIDS?
PHPIDS (PHP Intrusion Detection System) is an open source PHP Web Application Intrusion Detection System. It was written in March 2007. The main goal of PHPIDS is to give every PHP programmer the ability of finding intrusion data coming from client to php web application. Speed up PHP application development by reducing the amount of time and money needed to spend on application security.

(https://en.wikipedia.org/wiki/PHPIDS)

23)

DVWA also comes with a Web Application Firewall (WAF) called PHP-IDS. So PHPIDS is a WAF.

(Benim Not - https://blog.g0tmi1k.com/dvwa/index/)

24)

Analiz edilecek log'lar arasında ne tip özellikte kayıt arandığı en başta belirlenmelidir. Yani linux araçlarıyla log analizi yapabilmek için öncelikle log içerisinde ne arandığına dair teknik bir ölçüt belirlenmelidir. Örneğin Wordpress uygulamasının yüklü olduğu bir sunucuda /wp-admin.php ya da /wp-login.php gibi dizinlere brute force denemeleri yapılabilir. Log analiz araçlarına saldırı imzası olarak /wp-admin.php ya da /wp-login.php dersek, yani log kayıtlarında aranacak kelime olarak bunları belirtirsek saldırı yapanların bir kısmını belirlemiş oluruz.

(Page 39)

25)

Aşağıdaki komut bütünüyle beraber apache log'larında hangi IP adresi kaç adet bağlantı gerçekleştirmiş sonucunu verir:

```
> cat cyber_access_log | awk -F " " ' {print $1} ' | sort -n | uniq -c | sort -nr | head
```

(awk) -F ' ' : field separator'ı olarak boşluk belirlenir
(sort) -n   : numeric sort yapmaya yarar
(uniq) -c   : Tekrar eden satırların önüne kaçar defa tekrar ettiği sayısını koyar.
(sort) -nr   : Reverse olarak (tersten) numeric sıralama yapar.

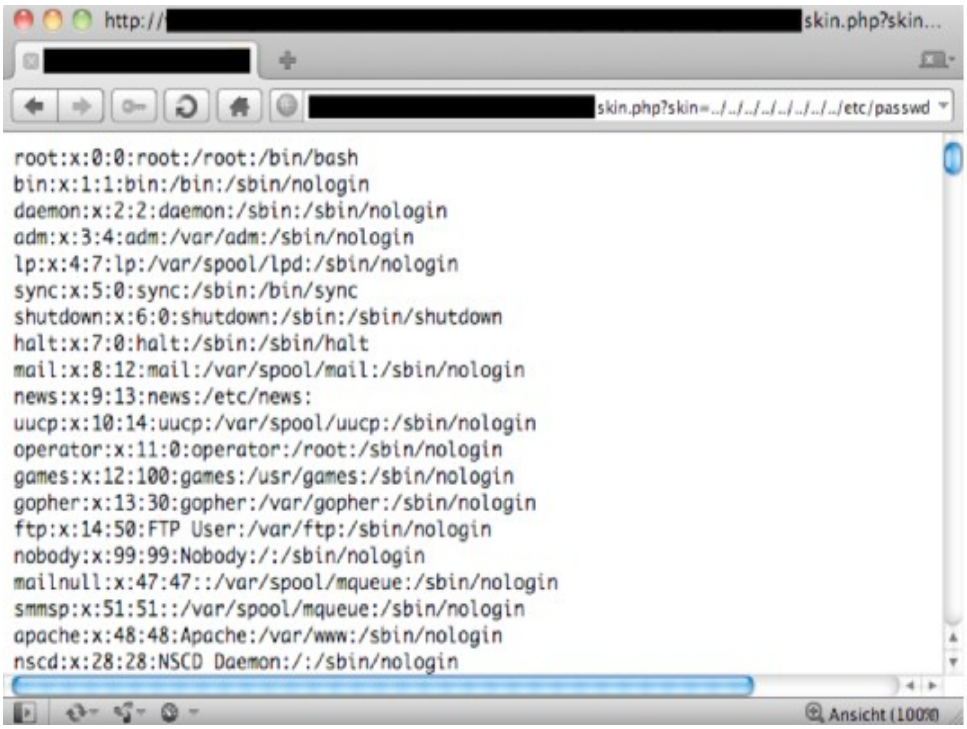
Output:

```
3556 9.6.2.2
1527 9.2.4.1
1142 1.1.2.8
1055 193.2.2.1
1046 9.1.2.1
```

Çıktıdaki ilk sütun ikinci sütunda yer alan IP adreslerinin log dosyasında kaçar defa tekrar ettiği bilgisini tutar. İkinci sutün da ilgili IP adresini tutar.
Yukarıdaki çıktının oluşum sürecini açıklayacak olursak cyber_access_log içerisinde sıralı kayıtlardan hepsinin ilk sütunları awk ile seçilir. İlk sütun field'ında web uygulamasına bağlanan IP'ler yer alır. Daha sonra sort -n ile bu IP'ler numeric olarak sıralanır. Daha sonra bu IP adresleri uniq'leştirilir. -c parametresi ile de her uniq IP adresinin kaçar defa tekrar ettiği bulunur ve ilgili IP adresinin başına konur. Daha sonra sort -nr ile bu sefer ters numeric bir sıralama yapılır ve en nihayetinde head komutu ile de elde edilen çıktının ilk 10 satırı ekrana yazdırılır.
(Page 41)

26)

Log Kayıtları Üzerinden LFI/RFI Saldırılarını Bulma
Web üzerinden gerçekleştirilebilecek önemli saldırı yöntemlerinden birisi de web uygulamasının yer aldığı sunucudaki dosyalardan birini okuyabilmektir. Buna Local File Inclusion derler.



Yukarıda /etc/passwd dosyasının LFI saldırısı ile okunduğunu görmektesiniz. Web üzerinden gerçekleştirilen LFI ya da RFI gibi saldırıları log'da yakalayabilmek için saldırı kodunda kullanılan ../ ya da ..\ gibi özel ifadeleri aratmak yeterli olacaktır. Fakat yine burada hatırlanması gereken önemli nokta bu aratılan karakterler GET isteği üzerinden yapılan saldırıları size buldurabilecektir.

Aşağıda LFI saldırısının tespit edilebildiği örnek log kayıtlarını görmektesiniz.

13.22.1.129 – - [01/Dec/2010:02:20:55 +0200] "GET /imprimer.asp?no=/../../../../../../../../../etc/passwd|44|80040e14|[Microsoft][ODBC_SQL_Server_Driver][SQL_Server]Line_1:_Incorrect_syntax_near_'/'. HTTP/1.1" 404 210 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3"

19.22.1.1 – - [01/Dec/2010:02:20:55 +0200] "GET /mailview.cgi?cmd=view&fldrname=inbox&select=1&html=../../../../../../etc/passwd HTTP/1.1" 404 210 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3"

13.22.1.19 – - [01/Dec/2010:02:20:55 +0200] "GET /modif_infos.asp?n=../../../../../../../../../etc/passwd%00 HTTP/1.1" 404 213 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3"

13.22.1.19 – - [01/Dec/2010:02:20:55 +0200] "GET /modif_infos.asp?n=/../boot.ini HTTP/1.1" 404 213 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3"

13.22.1.19 – - [01/Dec/2010:02:20:55 +0200] "GET /modif_infos.asp?n=/../../../../../../../../../etc/passwd HTTP/1.1" 404 213 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3"

13.22.1.19 – - [01/Dec/2010:02:20:56 +0200] "GET /pm/lib.inc.php HTTP/1.1" 404 212 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3"

13.22.1.19 – - [01/Dec/2010:02:20:56 +0200] "GET /productcart/pc/Custva.asp?|-|0|404_Object_Not_Found HTTP/1.1" 404 223 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3"

13.22.1.19 – - [01/Dec/2010:02:20:56 +0200] "GET /ProductCart/pc/msg.asp?|-|0|404_Object_Not_Found HTTP/1.1" 404 220 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3"

13.22.1.19 – - [01/Dec/2010:02:20:56 +0200] "GET /rubrique.asp?no=../../../../../../../../../etc/passwd%00|55|80040e14|[Microsoft][ODBC_SQL_Server_Driver][SQL_Server]Line_1:_Incorrect_syntax_near_'/'. HTTP/1.1" 404 210 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3"

13.22.1.19 – - [01/Dec/2010:02:20:56 +0200] "GET /rubrique.asp?no=/../boot.ini|55|80040e14|[Microsoft][ODBC_SQL_Server_Driver][SQL_Server]Line_1:_Incorrect_syntax_near_'/'. HTTP/1.1" 404 210 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3"

Bu kayıtlarda önemli olan saldırı teşebbüslerine karşılık web sunucunun döndüğü cevaptır. Web sunucu 404 değil de 300 veya 200'lü bir cevap dönmüşse saldırının başarılı olmuş olma ihtimali vardır. 404 alınıyorsa bu saldırganın denediği atakların başarılı olmadığı, sunucu tarafında istenen dosyanın bulunamadığı anlamına gelir.

13.22.1.19 – - [01/Dec/2010:02:20:56 +0200] "GET /rubrique.asp?no=/../../../etc/passwd|55|80040e14|[Microsoft][ODBC_SQL_Server_Driver][SQL_Server]Line_1:_Incorrect_syntax_near_'/'. HTTP/1.1" 404 210 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3"

(Page 42-45)

26)

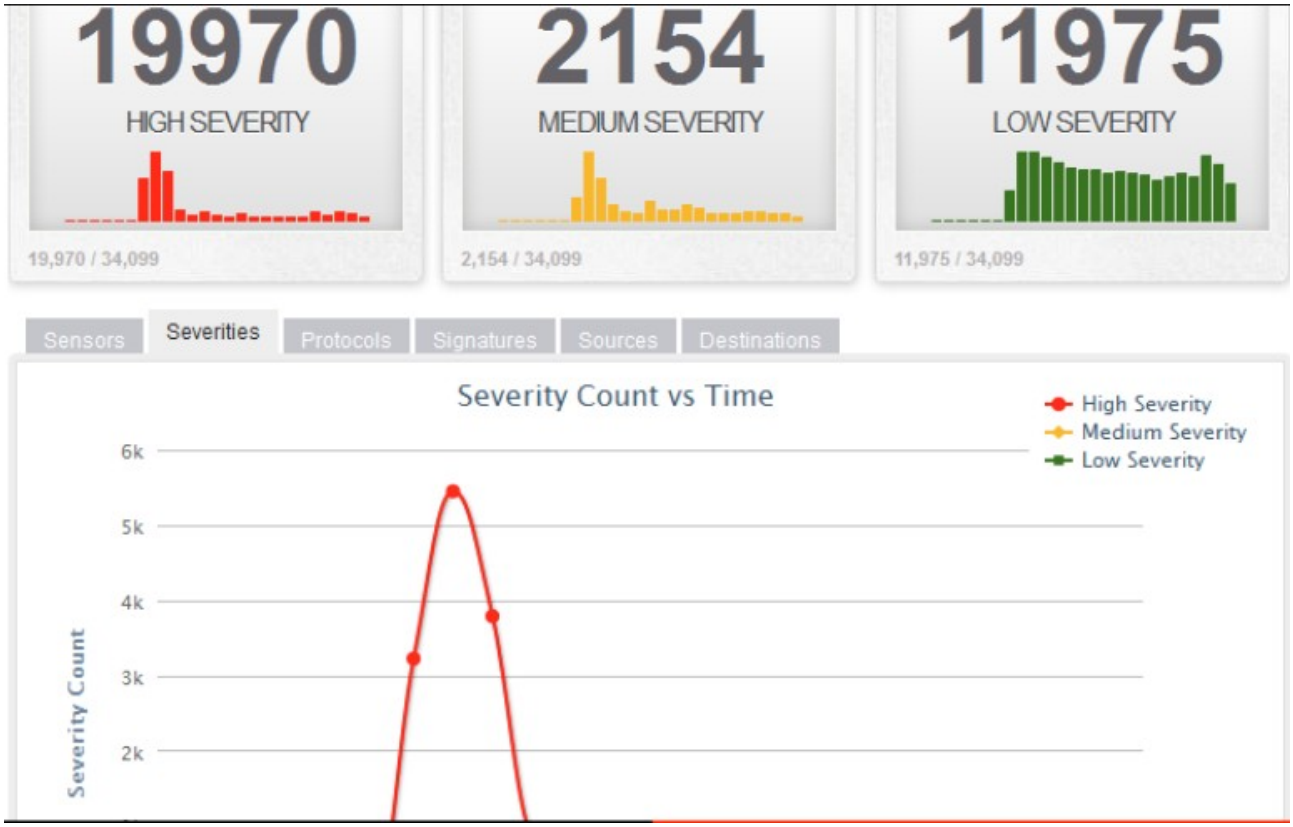
Ağ Trafiğini Log'lama Üzerine Yazılımlar

- Netwitness
- Ngrep
- Xplico
- Tcpdump
- Wireshark

(Page 46)

27)

Ağ üzerinden saldırı tespiti için Snort'un kullanımına dair bir ekran görüntüsü



(Page 49)