ADUR[IT] MAGAZINE

issue 1 - March 2009

TABLE OF CONTENTS

Penetration Testing Presentation

Autor : Mjeed

ال Penetration Testing هو محاولة اثبات أن نظاما معينا غير آمن ، بمعنى آخر محاولة اختراق نظام معين عن طريق الوصول الى معلومات لا يجب الوصول اليها من طرف الغير المصرح لهم ، أو القدرة على العبث بمعلومات معينة قبل اتمام ارسالها الى الطرف الاخر ، أو القدرة على الاطاحة بنظام معين بجعل الوصول اليه غير ممكن لفترة زمنية معينة. عادة ما يقوم باختبار الاختراق لمنظمة معينة جهة أخرى Third-Party عن طريق عقد مبرم بين الجهتين له بنوده المختلفة. كذلك يقوم بعض أفراد المنظمة أحيانا بالقيام بهذه العملية بشكل دوري للتأكد من أنظمة الحماية لديهم وتقديم تقرير أسبوعي-شهري-سنوي عن الحالة الأمنية للأنظمة.

ما أهمية اختبار الاختراق؟ ولم يجب القيام به؟

في الحقيقة هناك أسباب كثيرة ومن أهمها ، أن أدوات الاختراق أصبح في متناول الجميع ومعظمها لا يحتاج الى خبير أو مبرمج ، بل أن معظمها اتجه الى التتمتة في التشغيل مما قد لا يتطلب من المستخدم أكثر من ضغطة زر . كذلك معظم الشركات أصبحت توفر خدمات للعملاء من خارج نطاق المنظمة كتطبيقات الويب مما يجعلها عرضة للمتطفلين. وكما يقال في المثل ، الوقاية خير من العلاج ، فعندما يكتشف خلل أمني معين قبل حدوثه فعليا ، فحمايته من هجوم فعلي مستقبلا سيوفر على المنظمة تكبد خسائر جسيمة.

الخطوات المتبعة في عملية اختبار الاختراق :

قبل الشروع في عملية الـ PT هناك بعض الخطوات التي يجب المرور بها تدريجيا ويمكن أن أقسم العملية كاملة الى 4 مراحل وهي:

High-Level Assesment تقييم عام لسياسات المنظمة ، اجرائاتها ، ومعاييرها. وهذا الجزء هو جزء نظري بحت ولكنه مهم جدا لأنه المنطلق لاختيار نقطة الانطلاق للمراحل القادمة.

Network Evaluation وهو تقييم لشبكة المنظمة المراد اختبارها وهنا تبجأ جرعات الجانب العملي. حيث تتم محاولة معرفة طريقة تصميم الشبكة وكيفية ربطها بالمصادر المختلفة.

Low-Level Assesment وهنا نكمل ما اتممناه في المرحلة الأولى ولكن بشكل عملي بحت حيث يتم في هذه المرحلة مسح الشبكة ومحاولة استخراج أكبر قدر من المعلومات عن المصادر أو الأصول المختلفة.

Penetration Testing نصل الان الى مرادنا وهو القيام بعملية محاولة اختراق الأنظمة فعليا عن طريق ما تم استنتاجه من المراحل السابقة .

في الـ Penetration Testing عادة ما تتم العملية بدون معرفة أي معلومات عن الشركة ، أي كأنك لا تعرفهم وتحاول أن تصل الى مبتغاك ويسمى هذا بالـ black box approach ، وهو الغالب. في حين أن المنطلق الآخر وهو الـ white box box approach يتم فيه اعطاء كافة المعلومات التي يطلبها الـ P Tester . ومن المنطقي أن الغالبية تسلك الطريقة الأولى حيث أنه أكثر محاكاة لما قد يحصل في الواقع.

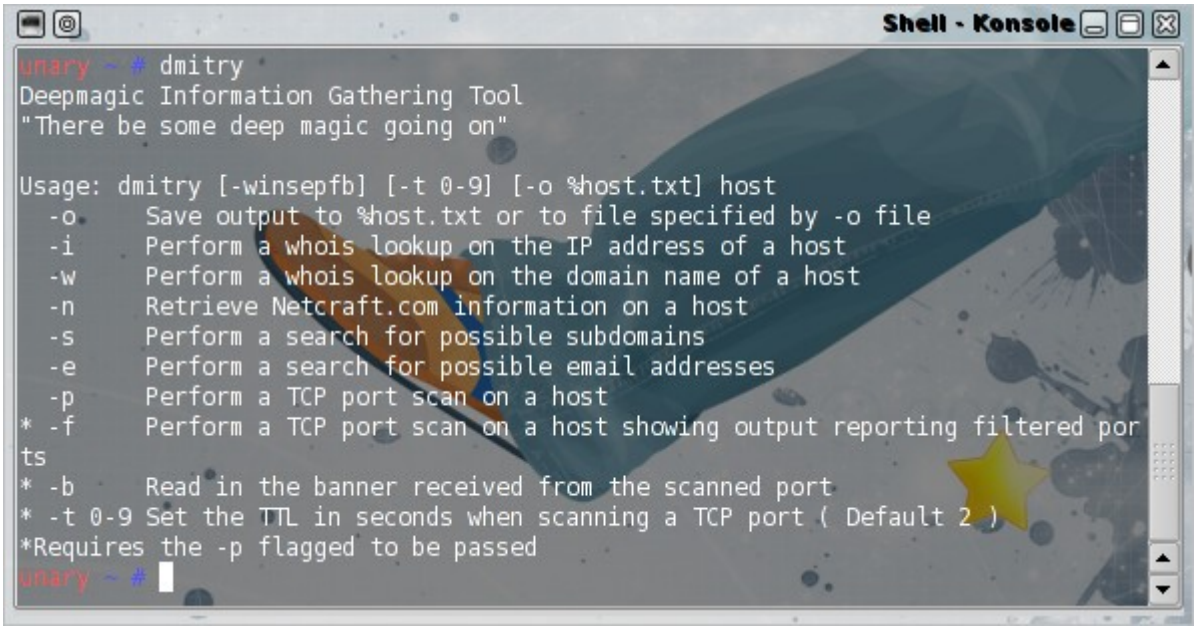
وما يميز تجارب اختبار الاختراق عن الهجومات الحقيقية انه لايتم تنيفيذ بعض المراحل المتعلقة بالهجومات الحقيقية اد ليس هناك تتبيت للبرمجيات الخبيثة من تروجونات او فيروسات او ديدان او RootKit على الهذف فهذا لا يعقل , و لا يتم التغيير في البيانات او تزويرها او حذفها - حذف ملفات Log الى غير ذلك ... - .

في المقال السابق طرح الأخ عبد المجيد المحارب نظرة عامة تعريفة جد مفيذة عن مجال اختبار الاختراق , كما تطرق الى السيناريو المتبع خلال عملية ال Penetration Testing .
لكن يبقى السؤال المطروح كيف يمكنني ان اطور مهاراتي في هذا المجال ؟
تتوفر على شبكة الانترنت العديد من الأدوات الخاصة بهذا المجال و هي متوفر للجميع سواء كان الشخص يريد التعلم انطلاقا من استخدام تلك الأدوات في حدود شبكته طبعا او كان الشخص دو عقلية تخريبية و هدا الموجود بالكثرة للاسف . و بما ان عملية جمع و توفير تلك الأدوات شبه صعب بالنسبة للمهتمين و المتخصصين في هذا المجال قامت عدة فرق بارزة في هذا الميدان بعمل توزيعات تشمل جميع الأدوات التي يحتاجها ال Pentesters في القيام بعملهم .
لكن الأمرالثاني هو كيف يمكن ان استغل هذه التوزيعات في تطوير مهاراتي و بشكل قانوني لا اتعدى فيه على ملكيات الأخرين , الحل متوفر و بسيط , و ذلك عن طريق Pentest Labs حيت هناك العديد من المشاريع تدعم هذه الفكرة كل ما عليك هو توفير محيط للعمل و من لا يملك الامكانيات يقوم بتنصيب ال VirtualBox او VMware و اعداده بالشكل اللازم و بدأ العمل بالاضافة الى تحميل بعض المشاريع التي توفر Pentset Labs كتوزيعات فريق De-ICE و هي توزيعات مخصصة لذلك و تختلف فيها مستويات الصعوبة مما يمكن المتدرب من تطوير مهاراته بالشكل المطلوب .
و نصيحة مني لكل من يريد الدخول الى مجال ال Penetration Testing فهذا المجال ليس فقط معرفة التعامل مع الأدوات و استغلالها بالشكل الصحيح بل يجب الاهتمام بالجانب النظري في كل شيء تتعلمه من شبكات و برمجة ....

http://www.de-ice.net/
http://en.wikipedia.org/wiki/Penetration_test

Information Gathering : Dmitry

Autor : unary



أداة تندرج تحت تصنيف ال Information Gathering او تفيدنا في جمع المعلومات عن الهذف حيت توفر العديد من الخصائص المهمة لعمل ذلك و يمكن استخدامها في عمل Whois على عنوان او Ip معين بالاضافة الى امكانية عمل Scan على المنافذ مع امكانية قراءة ال Banners المرسلة من طرف هذه الأخيرة ....

متال لاستخدام الأداة :

```
unary ~ # dmitry -ise youtube.com -o enum
Deepmagic Information Gathering Tool
"There be some deep magic going on"

Writing output to 'enum.txt'

HostIP:208.65.153.238
HostName:youtube.com

Gathered Inet-whois information for 208.65.153.238
---------------------------------
```

OrgName: YouTube, Inc.
OrgID: YOUTU
Address: 71 E Third Ave
Address: 2nd Floor
City: San Mateo
StateProv: CA
PostalCode: 94401
Country: US

NetRange: 208.65.152.0 - 208.65.155.255
CIDR: 208.65.152.0/22
NetName: YOUTUBE
NetHandle: NET-208-65-152-0-1
Parent: NET-208-0-0-0-0
NetType: Direct Assignment
NameServer: DNS1.SJL.YOUTUBE.COM
NameServer: DNS2.SJL.YOUTUBE.COM
Comment:
RegDate: 2006-03-02
Updated: 2006-03-09

RTechHandle: NETWO1084-ARIN
RTechName: networkradbaccount
RTechPhone: +1-650-343-2960
RTechEmail: radb@youtube.com

OrgTechHandle: NETWO1084-ARIN
OrgTechName: networkradbaccount
OrgTechPhone: +1-650-343-2960
OrgTechEmail: radb@youtube.com

ARIN WHOIS database, last updated 2008-10-30 19:10
Enter ? for additional hints on searching ARIN's WHOIS database.

Gathered Subdomain information for youtube.com
———————————
Searching Google.com:80…
HostName:www.youtube.com
HostIP:208.117.236.69
HostName:img.youtube.com
HostIP:72.14.247.118
HostName:uk.youtube.com
HostIP:208.117.236.69
HostName:help.youtube.com
HostIP:74.125.77.103
HostName:ca.youtube.com
HostIP:208.117.236.69
HostName:in.youtube.com

```
HostIP:208.117.236.69
HostName:au.youtube.com
HostIP:208.117.236.69
HostName:ie.youtube.com
HostIP:208.117.236.69
HostName:m.youtube.com
HostIP:208.65.153.240
HostName:nz.youtube.com
HostIP:208.117.236.69
Searching Altavista.com:80…
Found 10 possible subdomain(s) for host youtube.com, Searched 0 pages containing 0 results

Gathered E-Mail information for youtube.com
_____________________
Searching Google.com:80…
Searching Altavista.com:80…
Found 0 E-Mail(s) for host youtube.com, Searched 0 pages containing 0 results
```

Network Mapping : Nmap intro

Autor : unary

Nmap اختصار ل Network Maping أداة عنية عن التعريف لا يستغني عنها كل المهتمين بمجال الشبكات من مدراء و محللين أو غيرهم من ال Pentesters حيث يوفر العديد من المزايا و الخصائص تمكن من عمل فحص جهاز معين ضمن شبكة أو جميع الأجهزة الموجودة على نطاق شبكة معينة بالكامل و عرض لقائمة المنافذ المفتوحة بالاضافة الى امكانية تحديد اصدار الخدمات التس تستغل هذه المنافذ كما يعتمد nmap ايضا ال Os Fingerprinting بغية تحديد نظام التشغيل المستخدم في الجهاز الهذف و تمكننا هذه الخطوة من استخلاص معلومات عن أنظمة التشغيل و التي ستفيدنا في حالة وجود ثغرات خاصة بهذا النظام و يقوم مبدأ ال OS Fingerprint على ارسال مجموعة من الحزم الى الجهاز المستهذف و مراقبة الحزم الواردة و انطلاقا من هذه الأخيرة نستنتج ال fingerprint او بصمة نظام التشغيل الموجود على الجهاز و اداة ال Nmap توفر قاعدة بيانات للعديد من انظمة لتشغيل المتوفرة حيت تعمل الأداة على ارسال مجموعة من الحزم الى الجهاز الهذف و تعمل هلى مقارنة الحزم الصادرة من هذا الأخير و تحليلها مقارنة مع قاعدة بياناتها و من تم تحدد نوع نظام التشغيل المستعمل في الجهاز المستهذف

وسأتطرق في هذا المقال لبعض الخواص التي تفيذنا في عملية الفحص دون التعمق في ذلك أو التطرق الى الجانب النظري منها .
حسنا نبدأ بأول متال حيث سنقوم باستغلال ال Ping Scan لمعرفة هل الجهاز المراد فحصه on live أم لا و للقيام بذلك ننفذ الأمر التالي :

```
nmap –sP target
```

و لعرض المنافذ التي تستخدم بروتوكول TCP نستخدم TCP SYN Scan عن طريق تنفيذ الأمر التالي :

```
nmap –sS target
```

و لعمل فحص على المنافذ التي تستخدم بروتوكول UDP نستخدم UDP Scan عن طريق تنفيذ الامر التالي :

```
nmap –sU target
```

و لعمل os fingerprint مع تحديد اصدار الخدمات المتوفرة على الجهاز الذي نقوم بفحصه نستخدم الامر التالي :

```
nmap –sSV –O target
```

أما لتحديد اصدار خدمة واحدة وذلك بتعيين المنفذ الذي تستخدمه هذه الخدمة نقوم بتنفيذ الامر كالتالي و نأخد كمتال خدمة FTP

```
nmap –sV –p 21 target
```

كما ان nmap يمكننا ايضا من تزوير مصدر الفحص عن طريق تحديد IP مزور Spoofed IP يستغله nmap في عملية الفحص بالاضافة الى ال IP الخاص بنا حيت لن يتم التعرف على المصدر الحقيقي لعملية الفحص و يمكن تنفيذ الامر كالتالي :

```
nmap –S spoofed_IP –e eth* -P0 target
```

حيت تستخدم مكان * رقم network interface .

يمكن ايضا استخدام ك Packet Trace لتتبع مسار ال Packets باستخدام الامر :

```
nmap –packet-trace target
```

و لحفظ ما قمت به ف nmap يوفر لك ذلك اما عن طريق ملف عادي او تصدير النتائج على شكل ملف xml .
بالنسبة للخيار الأول ننفذ الامر التالي :

```
nmap –oN result target
```

و بالنسبة للخيار الثاني :

```
nmap –oX result.xml target
```

نكتفي بهذا القدر عن nmap حيت لا يمكن حصر شرح الخواص التي توفرها لنا هذه الأداة في مقال واحد و هناك كتب كاملة تتحدث عن الأداة بالتفصيل الممل و للتعرف اكثر عن طرق استخدام الأداة راجع صفحات ال man الخاصة بالأداة .

```
man nmap
```

http://www.mor-pah.net/index.php?file=projects/dmitry
http://nmap.org/

Scanners : Nikto

Autor : unary

أداة لايجاد الملفات الافتراضية , البرامج القديمة , الملفات الغير المحمية و اعدادت خوادم الويب و فحص تطبيقات CGI .

و هي من الادوات الفعالة و المجانية لتقييم و فحص امن خوادم الويب . لكن في حالة الاستعمال الخاطيء قد تتسبب في عمل crash للسيرفر لذلك يعد استخدام متل هذه البرمجيات غير قانوني الا في حالة كان السيرفر المراد فحصه في ملكية المستخدم .

هنا سأعرض ابسط طرق التعامل مع هذه الاداة :

- الخاصية h حيت يكون المنفذ الافتراضي هو 80 :

```
nikto -h host_ip
```

- فحص عن طريق منفذ محدد :

```
nikto -h host_ip -p port
```

- فحص عن طريق مجال من المنافذ :

```
nikto -h host_ip -p start_port-end_port
```

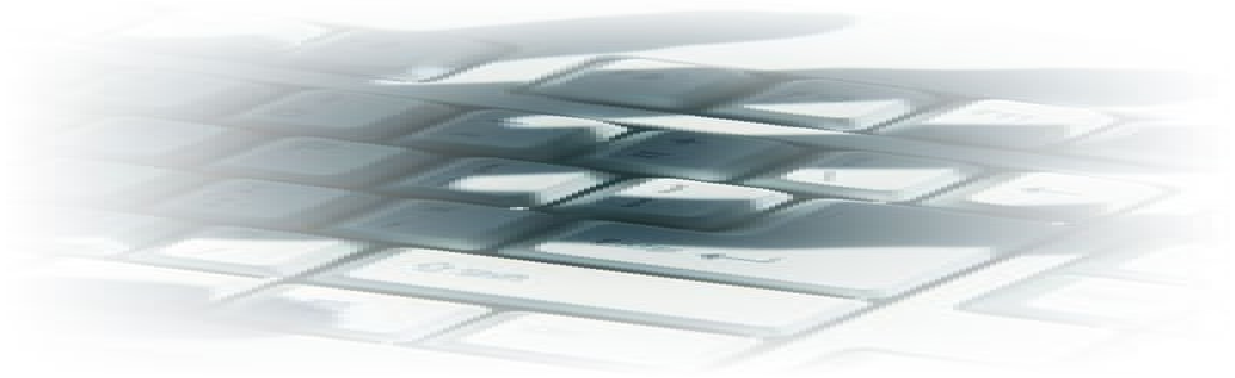
- فحص عن طريق عدة منافذ محددة :

```
nikto -h host_ip -p port1,port2….
```

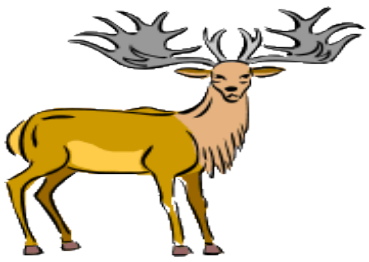
- للمزيد عن الاداة راجع التالي:

```
man nikto

nikto –help
```



Web Application Vulnerability Scanner : Wapiti



أداة تم تطويرها بلغة ال python , و هي من اداوات ال web application vulnerability scanner تفيدنا الأداة من اختبار سلامة تطبيقات الويب و اكتشاف نقاط الضعف فيها مما بمكننا من تحسينها و تصحيح الأخطاء البرمجية التي ادت الى هذا الضعف .
و الأداة يمكن ان تكتشف انواع الثغرات التالية :

```
File Handling Errors (Local and remote include/require, fopen, readfile…)
Database Injection (PHP/JSP/ASP SQL Injections and XPath Injections)
XSS (Cross Site Scripting) Injection
LDAP Injection
Command Execution detection (eval(), system(), passtru()…)
CRLF Injection (HTTP Response Splitting, session fixation…)
```

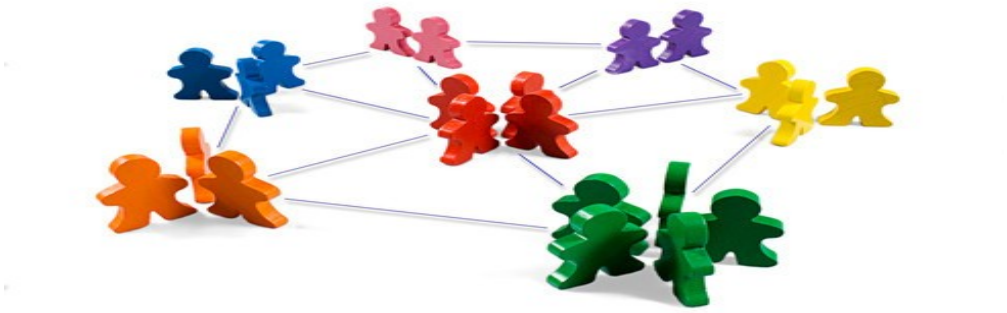
و بعد هذه النبذة البسيطة عن الأداة دعنا نتطرق لبعض الخصائص التي توفرها , و من ابسط الطرق في استخدام الأداة يكفي تنفيذ الأمر التالي لبدء عملية الاختبار :

```
wapiti.py target
```

اما ان اردت ان تكون متخفيا خلال عملية الاختبار فالأداة توفر لنا ذلك حيث بمكن استخدام proxy اثناء عملية ال scan

و توفر الأداة ذلك عن طريق الخاصية -p و بالتالي يصبح الامر كالتالي :

```
wapiti.py target -p proxy:port
```



Network Vulnerability Scanner : SAINT

Autor : unary

Saint : احدى المشاريـع المتخصـصة فـي ال Network Vulnerability Scanning و هو من أدوات عمل ال Penetartion Testing حيث يمكن من اختبار مستوى الأمان في الشبكة التي نريد فحصها كما يساعد في اعداد التقارير الخاصة بال Pentest حيث لابد من عمل اعداد تقرير عن حالة الشبكة عند عملية اختبار الاختراق .
برنامـج Saint هـو الخليفـة للبرنامـج Satan اختصـار ل Security Administrator Tool For Analyzing Networks و Saint غيـر مجانـي لكـن يمكـن تحميله و تجربتـه بعـد التسـجيل بالموقـع الرسـمي الخاص بهذا البرنامج و الحصول على ال key بعد تحديد رقمي الجهازين المراد فحصهما و البرنامج يشتغل فقط على منصة اللينكس و ال FreeBSD و Solaris و ال MAC OS X .

http://www.saintcorporation.com

Network Protocol analyzer : Tshark

Autor : Djekmani

TShark هو اصدار من الـ Wireshark يشتغل على سطر الاوامر وبدون واجهة رسومية صمم لإلتقاط وقراءة Packet وتجد بـه كل خيارات الموجودة في wireshark , يمكن لك بهدا البرنامج قراءة كل البيانات المارة من الشبكة وتحليلهـــا كمـــا يمكـــن اضهار النتائج فـــي ملف وعدم اظهار النتائج على interactive terminal ان كنت غير فاضي لمراقبتها في دلك الوقت يمكن لك ان تراجعها فيما بعد . لندرس بعض الخيارات التي يتيحها لنا هدا البرنامج

اولا لتتبيت TShark في debian وعائلتها :

sudo apt-get install tshark

لتتبيته على Redhat/Fedora وعائلتهم :

yum install tshark

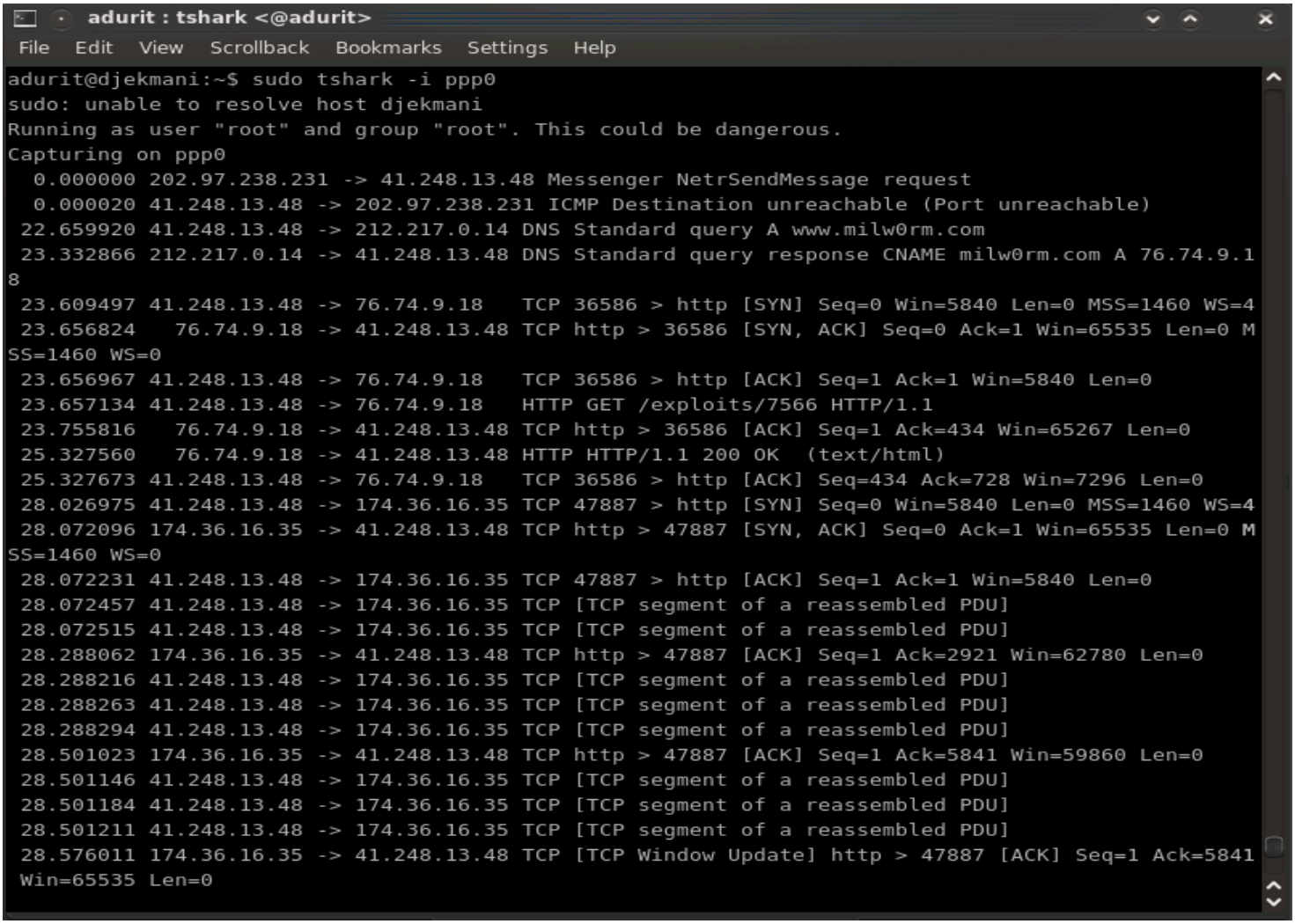
•بعد تتبيت البرنامج بنجاح نشغله :

```
tshark -i <Interface>
```

الـ option -i تعني تحديد نوع الشبكة التي سوف تعمل عليها متال :

```
adurit@djekmani:~$  tshark -i ppp0
```

هده صورة من جهازي تبين التقاط الاداة للـ Packets المارة



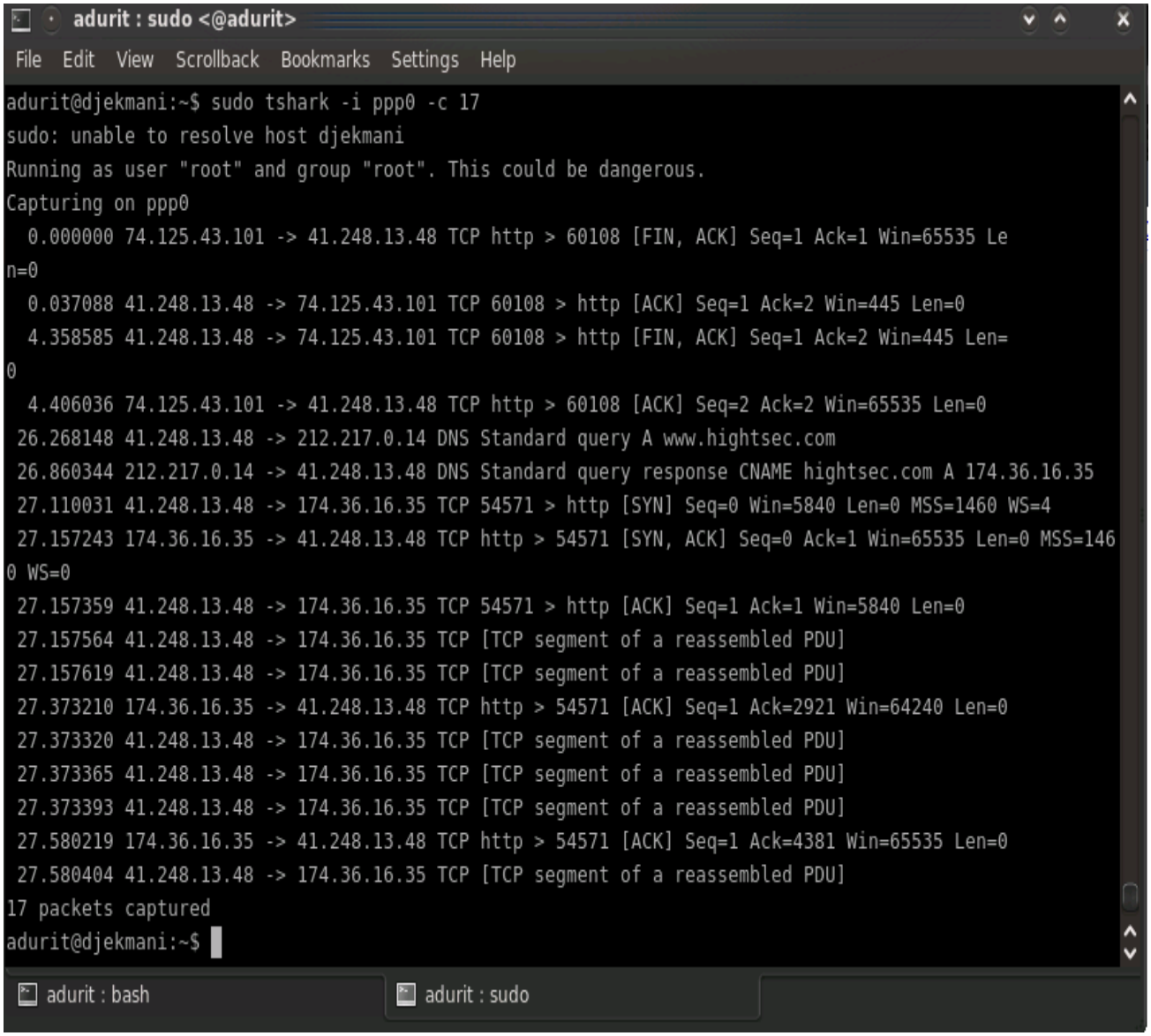
•لتحديد عدد الـ packet التي تلتقط تم يقف الـ capturing

```
tshark -c < packet count>
```

ناخد على هده الـ option مثال حي

```
adurit@djekmani:~$ tshark -c 17 -i ppp0
```

وهده صورة من جهازي توضح بالالوان



- يمكن لك وضع النتائج في ملف ومن بعد تقرء النتائج وتحللها :

```
adurit@djekmani:~$ sudo tshark -f -c 17 -i ppp0 >> adurit.dj
```

سوف تكون plain text يمكن ان تجد صعوبة في قراءة النتائج لدلك فهنالك Option تعمل encode للـ packet ويتم عمل decode بواسطة ادوات التي تدعم libpcap وهدا option يعطيك لائحة من هده البرامج سوف نراها لاحقا

```
tshark -F
```

- نكمل كيفيت ادخال النتائج في outpute file

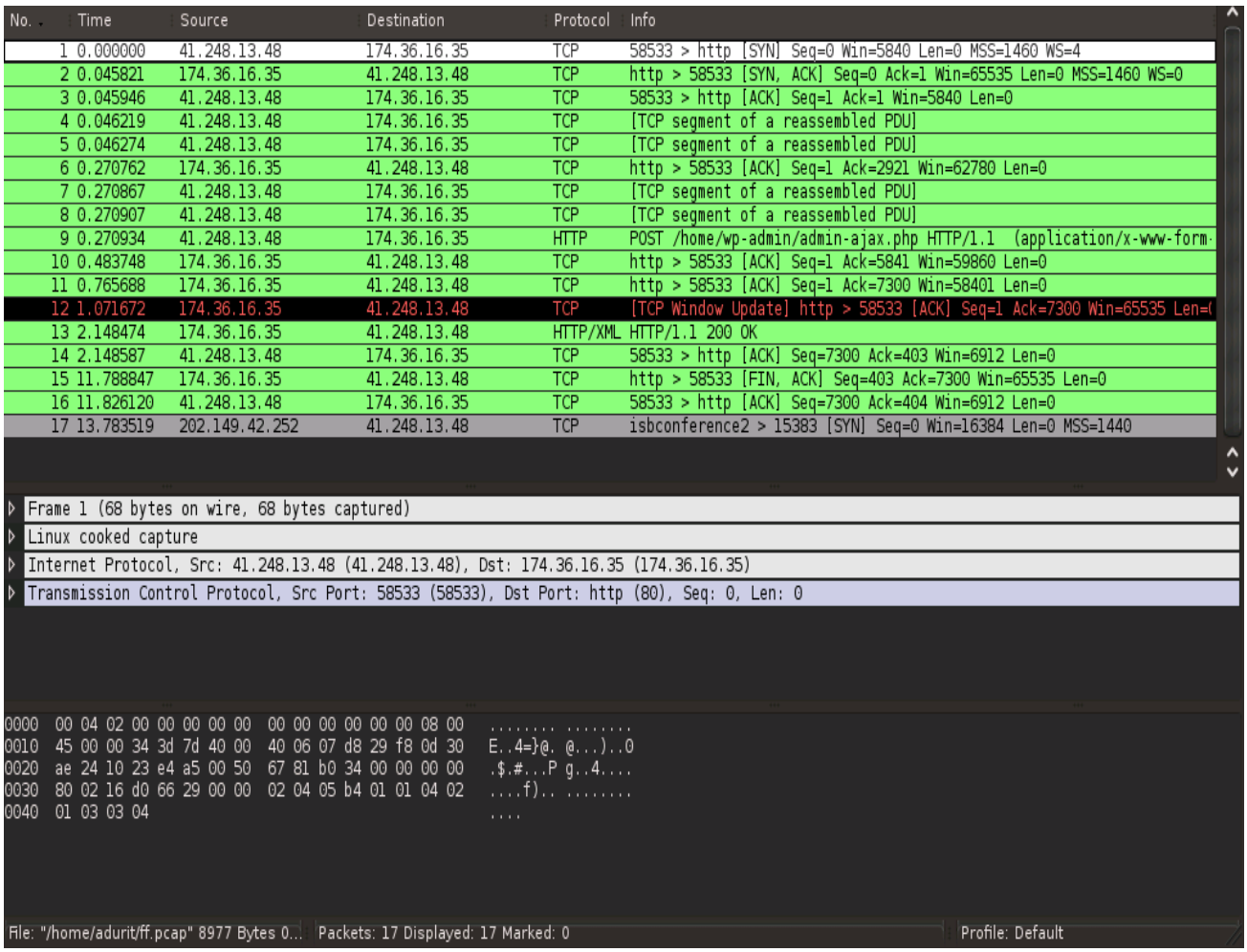
```
adurit@djekmani:~$ tshark -c 17 -i ppp0 -w dj.pcap
```

ان اردت فتح ملف dj.pcap سوف نستعمل احد البرامج التي قلنا عليها وهده بعضها :

```
libpcap - Wireshark/tcpdump/… - libpcap
nseclibpcap - Wireshark - nanosecond libpcap
modlibpcap - Modified tcpdump - libpcap
nokialibpcap - Nokia tcpdump - libpcap
rh6_1libpcap - RedHat 6.1 tcpdump - libpcap
suse6_3libpcap - SuSE 6.3 tcpdump - libpcap
5views - Accellent 5Views capture
dct2000 - Catapult DCT2000 trace (.out format)
nettl - HP-UX nettl trace
netmon1 - Microsoft NetMon 1.x
netmon2 - Microsoft NetMon 2.x
ngsniffer - NA Sniffer (DOS)
ngwsniffer_1_1 - NA Sniffer (Windows) 1.1
ngwsniffer_2_0 - NA Sniffer (Windows) 2.00x
niobserverv9 - Network Instruments Observer (V9)
lanalyzer - Novell LANalyzer
snoop - Sun snoop
rf5 - Tektronix K12xx 32-bit .rf5 format
visual - Visual Networks traffic capture
k12text - K12 text file
commview - TamoSoft CommView
pcapng - Wireshark - pcapng (experimental)
```

سوف نستخدم منها wireshark لان موضوعنا كله shark الله ينجينا وينجيكم lol

نفتح الملف باداة الـ wireshark بيظهرلك النتيجة على واجهته , وهده صورة ملتقطة من جهازي توضح ما قلته



هده مقدمة بسيطة عن اداة tshark وإمكانياتها الهائلة في Dump and analyze network traffic ونترك لك المجال في التعرف اكثر واكثر عن الادات :

```
tshark -help

man tshark
```

Packet Injection And Analysis Tool : Packit

Autor : unary

Packit : أداة تندرج ضمن قائمة أدوات ال Spoofing و تفيذ كل مهتم بدراسة الشبكات بالاضافة الى المتخصصين في هذا المجال او المهتمين به حيت تمكننا من عمل مراجعة للشبكات أي Network Auditing و تحليلها و تتميز بقدرتها على حقن و رصد و التلاعب بال Trafic حيت تمكننا من تزوير حزم تروتوكولات مختلفة TCP , UDP , ARP , RARP بالاضافة الى ال Ethernet Headers .
كما أن الأداة توفر مزيدا من الخصائص للمتخصصين في حماية الشبكات حيت تمكنهم من اختبار الجدران النارية Firewalls و اختبار أنظمة كشف الدخلاء و كل ما يخص ال TCP/IP Auditing .
و ما سأتطرق اليه في هذا المقال هو بعض الخصائص التي توفرها لنا الأداة فيما يخص كل من Packet Capture و Packet Injection و Packet Trace
و Packit يمنكن استغلالها في هذه الاوضاع ككل حيت توفر لنا الخاصية m اختيار الوضع الذي نردي الاشتغال عليه و الوضع الافتراضي للأداة هو Injection Mode :

Packet Capture Mode

توفر لنا Packit عدة امكانيات في عمل ذلك حيت يعمل هذا الوضع على التقاط جميع ال Packet المارة من ال Network Interface الذي قمنا بتحديده و eth0 هو الافتراضي في الأداة و هذا الوضع يعمل عمل ال sniffer و يمكن تنفيده كالتالي :

```
# packit –m cap
```

و هنا ستعمل Packit على التقاط جميع الترافيك المار من eth0 و عرضه لنا و هذا يصعب علينا تحليله لدا دعنا نحدد عدد ال packets و عمل فلترة على مستوى البروتوكول و الاحتفاظ بالناتج في ملف لدراسته و تحليله :

```
#packit –m cap –c 60 ‘tcp’ –w capture.txt
```

و لقراءة ناتج الملف نطبق الامر التالي :

```
#packit –m cap –r capture.txt
```

Packet Injection Mode

هذا الوضع مفيد جدا لكل مهتم بمجال حماية الشبكات من المتطفلين حيث يمكننا من دراسة و تحليل و اختبار الجدران النارية و أنظمة كشف الدخلاء لتطويرها و زيادة مقدار فعاليتها بالاضافة الى كل ما يخص ال TCP/IP Auditing على كل دعنا نتطرق الى بعض المزايا التي توفرها لنا Packit في عمل ذلك حيث سنستغلها في ارسال مجموعة من ال Packets المزورة مع امكانية تزوير مصدر هذه الحزم أي عمل IP Spoofing ل IP الخاص بنا كما ان الأداة توفر كدالك امكانية عمل MAC Adress Spoofing . دعنا الان طريقة عمل ذلك :

```
#packit –d target_ip –s spoofed_ip –S 21 –D 80 –F S –c 25
```

الخاصية الاولى تمكن من تحديد الجهاز الذي نريد ارسال الحزم الخاصية الثانية تمكن من عمل تزوير لمصدر هذه الحزم و الخاصية الثالثة تمكن من تحديد Port source و الخاصية الرابعة تمكن من تحديد ال Port الذي نريد ارسال هذه الحزم اليه اي Distination Port و الخاصية الخامسة حددنا فيها ال Flags و هنا من نوع SYN و الخاصية الاخيرة تمكن من تحديد عدد الحزم المراد ارسالها . و هنا لم نحدد ال Mode الذي ستعمل عليه الاداة لاننا ندرس الوضع الافتراضي لها . كما يمكننا ارسال حموله اي Payload باستغلال ها الوضع يمكنك اكتشاف ذلك عند استخدامك للاداة .

Packet Trace Mode

يمكن عمل Packet Tracing باستخدام Packit كالتالي :

```
#packit –m trace –d target
```

نكتفي بهذا القدر عن هذه الأداة و اريد ان فقط ان اشير الى كيفية تنبيت الأداة حيت لابد من وجود كل من Libnet و Libpcap لتنبيت الأداة . و الامر سهل بالنسبة لمستخدمي التوزيعات المبنية على Debian يكفي تنفيد الامر التالي :

```
#apt-get install packit
```

و بالنسبة لاصحاب التوزيعات الأخرى فالتثبيت من ال source هو الحل و لاصحاب التوزيعات من عائلة REDHAT فهناك حزمة rpm متوفره في الموقع الرسمي للأداة .

http://www.packetfactory.net/projects/packit/

THC-Hydra : واحدة من اكبر الادوات في مجال تكسير كلمات السر تبع البروتوكولات وتعتمد على اخر تغرات brutforcing ..وتدعم عدد كبير من البروتوكولات التي تستطيع كسر كلمات السر الخاصة بمدرائهم ومن مميزاتها انها تعمل حتى في Secure Socket Layer يعني ممكن تعمل هجوم حتى على بروتوكولات المحمية متال(HTTPS ,FTPS..) تدعم الادات في الوقع الراهن :

TELNET, FTP, HTTP, HTTPS, HTTP-PROXY, SMB, SMBNT, MS-SQL, MYSQL, REXEC,RSH, RLOGIN, CVS, SNMP, SMTP-AUTH, SOCKS5, VNC, POP3, IMAP, NNTP, PCNFS,ICQ, SAP/R3, LDAP2, LDAP3, Postgres, Teamspeak, Cisco auth, Cisco enable,LDAP2, Cisco AAA (incorporated in telnet module).

هدا البرنامج يتبت صحة الكلام ان ممكن دخول الغير القانوني لاي خدمة مهما كانت محمية ويعني فرصة لاصحاب سكيورتي والباحتين على تطوير مهاراتهم واكتشاف كيفيت يتم تخطي حمايتهم

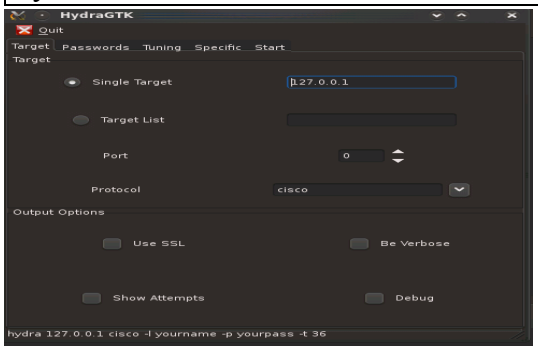
نشغلها بالاتي :

```
dj@bt2-Security hydra-5.4-src # hydra
```

راح يضهرلك USAGE+معلومات عن البرنامج .

وهذا متال لعمل attack لخدمة ال SSH :

```
hydra 196.217.100.9 ssh2 -s 22 -l root -P /root/x.txt -t 36
```



وتوجد العديد من الخاصيات اترككم تكتشفونها

فريق عمل البرنامج اعطى فرصة للمبتدئين ومحبين الرسميات فرصة وقدمو البرنامج بحلة رسومية بـ GTK .. واكسبوه رونق خاص

لتشغيلها اكتب في الترمنال :

```
dj@bt2-Security Hydra # xhydra
```

Network Logon Cracking : Medusa

Autor : unary

Medusa : Network LoGin Auditor

أداة من أدوات ال Logon Cracking و شبيهة الى حد ما بالأداة THC-Hydra و لكن تختلفان في بعض المواصفات و لكل واحدة مميزاتها عن الاخرى و تضم هذه الأدة حاليا وحدات للخدمات التالية :

- CVS

- FTP
- HTTP
- IMAP
- MS-SQL
- MySQL
- NCP (NetWare)
- NNTP
- PcAnywhere
- POP3
- PostgreSQL
- rexec
- rlogin
- rsh
- SMB
- SMTP (AUTH/VRFY)
- SNMP
- SSHv2
- SVN
- Telnet
- VmAuthd
- VNC

الأداتان على حد سواء لها نفس الخصائص حيت تسمح لك أن تحدد اسم مستخدم او قائمة أسماء مستخدمين ، و قائمة من كلمات السر للاختبار ضد بروتوكول معين يحدده المستخدم .

- نبذة عن استخدام الاداة :

- لرؤية الوحدات او ال Modules المتبتة ننفذ الامر :

```
root@unary:~# medusa -d
```

- متال 1 :

```
root@unary:~# medusa -h host -u unary -P password.txt -M ftp
```

في المتال التالي باختبار بروتوكول ال ftp عن طريق تحديد اسم مستخدم هنا unary و ملف به كلمات المرور المراد تجربتها و الخيار M يمكننا من تحديد البروتوكول المراد اختباره .

و ادا اردت تحديد قائمة باسماء المستخدمين نقوم بادراجهم في ملف نصي و ننفد الامر كالتالي :

```
root@unary:~# medusa -h host -U users.txt -P password.txt -M ftp
```

- متال 2 :

```
root@unary:~# medusa -h host -u unary -P password.txt -M ssh -n 2525
```

في المتال التالي عرفنا ان الخدمة ssh لا تشتغل على ال Port الافتراضي لذلك نقوم بتحديد ال port الذي تم تحديده للخدمة باستخدام الخيار n

نكتفي بهذين المتالين و نترك لكم البقية تكتشفونها بأنفسكم حيت يمكنك الاستعانة ب

```
meduza –help
```

http://freeworld.thc.org/thc-hydra/
http://www.foofus.net/jmk/meduza

SQL Injection : Introduction to SQL Column Truncation

Autor : Zigma

عند الحديث عن ثغرات تطبيقات الواب من المأكد أننا سوف نتوجه بالحديث عن ثغرات الحقن , لكننا نقوم بتهميش أخطاء أخرى تقع جراء عدم التحقق من المداخل و أهم هده الأخطاء تقع عندما يقوم المستخدم بإدخال قيم أكبر من اللازم أو الحد المسجل له , أظن أنني قمت بتوضيح الموضوع - ثغرات تقزيم الأعمدة :
قد أثيرت هذه القضية من قبل Stefan Esser .و تجد مواضيعه على مدونته suspekt.org .تتمثل ثغرات تقزيم الأعمدة في إدخال قيمة شبيهة بقيمة موجودة مسبقا لكن مع إختلاف جزئي يقوم به المخترق لإجبار التطبيق على أخذ القيمة و تخزينها في قاعدة البيانات.عندما نقوم بإدخال قيمة طويلة لقاعدة البيانات يقوم المايسكل بتقزيمها لو تخطت الحد المعلن لهاو هنا يكمن الخطر.لنفترض أننا نملك تطبيق ويب حيث يمكن للمستخدمين التسجيل (منتدى) + إسم المدير مثلا معروف و لنفترض أنه "Administrator" + لا يوجد حد لطول القيمة المدخلة عند عملية التسجيل .فلنفترض أن عمود القاعدة user .محدود ب 25 قيمة..إذا قام القرصان بمحاولة تسجيل عضو جديد بإسم Administrator ستفشل المحاولة نظرا لإستعمالنا دالة محددة تقوم بعمل بحث عن العضو و هل يوجد إسم بذلك الإسم و نطلق عليها دالة :

```
AlreadyRegistred()
```

و الان , فرظا لو أن القرصان قام بإدخال قيمة تفوق 25 خانة مثلا :

```
‘Administrator……x’
```

ستقوم دالتنا بالبحث عن هذا الإسم في قاعدة البيانات و طبعا لن تجده لأن عدد الخانات يتخطى ال 25 خانة و بالتالي التطبيق سيقوم بالقبول به في مرحلة أولى و إرساله إلى قاعدة البيانات على مستوى قاعدة البيانات سيقوم المايسكل بتقزيم الإسم نضرا لمحدودية العمود ب 25 قيمة و سيصبح

```
’Administrator ……‘
```

ملاحظة لا تعتبر النقط الموجودة في المتال تم وضعها لتبيين المتال مع التنسيق في المدونة

ونتيجة لدهاء هذا القرصان لنا الان مستخدمين لهم إسم Administrator.و هنا نرى الإشكال..نعود للتطبيقينا :
فرظا اننا عند عملية الدخول يقوم التطبيق بطلب الباسورد حسب إسم المستخدم و نظرا لكون القرصان حديثا ما قام بتسجيل الإسم المقزم في قاعدة البيانات نتيجة ال SELECT

ستقوم بإعطاء قيمته أولا و بالتالي كونه يعرف كلمة السر التي قد قام بتسجيل المدير المقزم بها سيتمكن من الولوج نحو التطبيق مع التمتع بصلاحيات الأدمن
هنا نختم هذا المقال بالقول بأن هذه الثغرات حديثة العهد و يتم إهمال خطورتها من قبل مبرمجي التطبيقات و ما إن تراءى أول مقال عنها برزت ثغرة في مدونة Wordpress
و لك بذلك أن تفهم قيمتها و كثرت إنتشارها .

Security : GreenSQL

Autor : unary

تشكل ثغرات قواعد البيانات تهديدا كبيرا خاصة في تطبيقات الويب و تعاني منها للأسف الكتير من المواقع على الشبكة العنكبوتية , حيت تمكن هذه النوعية من الثغرات من تنفيذ هجوم على قاعدة البيانات على شكل استعلامات تمكن المخترق من الحصول على معلومات من الممكن ان تكون جد مهمة كأرقام الاشتراكات في خدمة ما أو ارقام بطاقات الائتمان الى غير ذلك من المعلومات بالاضافة الى الوصول الى معلومات حساسة فانها تمكن المخترق من .... المهم هذا ليس محور موضوعنا .

بعد هذه المقدمة المملة ننتقل الى محور موضوعنا و هو عن GreenSQL , و هو مشروع مفتوح المصدر يمتل جدار ناري مخصص لقواعد البيانات لحمايتها من هجمات ال SQL Injection حيث يعمل بين الموقع و قاعدة البيانات و يحدد ماهية البيانات المتاح الوصول اليها و ماهي البيانات الممنوع الوصول اليها أي أنه يعمل نوع من فلترة للاستعلامات التي تنفذ على قاعدة البيانات .

و المشروع متوفر للتحميل لكل من CentOS , debian , Opensuse , fedora , ubuntu و FreeBSD و يتوفر على واجهة جميلة لادارته مع امكانيات عديدة في اعداده .

Linux : Introduction To Kernel Modules

Autor: Djekmani

احببت ان اخصص هدا المقال في تطوير الكرنيل و اساسيات كتابة Kernel Modules.اولا نعطي تعريف بسيط للموديل كرنل ال Kernel Modules تحديدا : هى مشابهه لل Device drivers وبتستخدم فى زيادة قدرات ال Kernel ومن اشكالها :

Device drivers File system drivers

```
System calls
Network drivers
```

خلال عملية ال Boot ال Kernel بتعمل Load لل Modules التالية تلقائيا :

```
/etc/modules.conf
/etc/conf.modules
/etc/modprobe.conf
```

ال Kernel Modules تقدر تضيفها وتحذفها Manually , ك تقدر تعمل Load او تعمل Unload لأى Modules فى اى وقت وال Kernel شغالة مش هتحتاج تعمـل Reboot غيـر ممكـن فـى حالة واحدة لو غيرت ال Kernel مثلا فـــي التدوينـــة دي راح نتعرف على بعـــض مبادئ بناء Kernel Modules يلزمـــك كاســـاسيات ان تكون لك خـــبرة فوق متوســـطة فـــي لغـــة C سوف ندرس في هدا الجزء :

1. Debug Kernel Mode
2. Loading And Unloading Module
3. Description Module
4. Passage 0f Parameters

نبدئ على بركة الله

Debug Kernel Mode

لكي تستطيع متابعة الدرس وتطبيقه سوف تحتاج لبعض الادوات غالبا ما تكون مدمجة مع توزيعتك .

```
printk() - dmesg - KGDB - KDB
```

هده الادوات التي سوف نحتاج في الطريقة التي سوف نستعمل . يوجد طرق اخرى بس ليست في موضوعنا ..

نمودج

```
int printk(const char *fmt, ...)
```

متال

```
printk("<1> Welcome To djekmani4ever's World !\n");
```

عديد من مستويات التنقيح تابتة في <linux/kernel.h>

```
#define KERN_EMERG    "<0>"  /* نظام غير مستعمل */
#define KERN_ALERT    "<1>"  /* انظار */
```

```
#define KERN_CRIT      "<2>"  /* حالة غير مستحبة */
#define KERN_ERR       "<3>"  /* حالة خطئ */
#define KERN_WARNING   "<4>"  /* رسالة تحديرية */
#define KERN_NOTICE    "<5>"  /* حالة عادية بس يعلمنا بالحالة */
#define KERN_INFO      "<6>"  /* معلومات */
#define KERN_DEBUG     "<7>"  /* رسالة بعد التنقيح */
```

كيفيت استخدامها في الموديل

```
printk(KERN_ALERT "Welcom To Djekmani4ever'S World !\n");
```

ملاحظة : الامر dmesg يمكنك من اضهار رسائل printk()

Loading And Unloading Module

Module يعني نقطة بداية ونقطة نهاية

```
int xxx(void) : نقطة البداية
void yyy(void): نقطة النهاية
```

ممكن تعمل بدل xxx و yyy الدي تبيه , لكي نقول ان لدينا module يجب على هتين الدالتين ان يكونا على شكل نقطة بداية ونقطة نهاية ...لدا نستخدم هدين macros

```
* module_init(xxx);
* module_exit(yyy);
```

في نهاية الموضوع الدالتان module_init و module_exit يستدعيان تلقائيا في عملية chargement and dechargement تبع الموديل ب insmod and rmmod

نشوف الان سورس صغيرة حلوة لموديل متكامل بعد دراسة شاملة لكيفيت تجهيز

```
#include <linux/module.h>
#include <linux/init.h>
static
int __init mon_module_init(void)
{
printk(KERN_DEBUG "Welcom 2 My BloG !n");
return 0;
}
```

```
static

void __exit mon_module_cleanup(void)
{
printk(KERN_DEBUG "Goodbye djekmani !n");
}
module_init(mon_module_init);
module_exit(mon_module_cleanup);
```

نجي الان نعمله Compile .. بعرف ان كل من يسمع الكلمة دي بيهرب .. بس هي عملية بسيطة

نستخدم الامر make

نعمله توجيه ب makefile

على هدا الشكل

```
obj-m += module.o

default:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules

clean:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

نعمل Chargement and déchargement بالتالي :

```
Djekmani@Bt3-security ~ # insmod ./module.ko
Djekmani@Bt3-security ~ # lsmod
Djekmani@Bt3-security ~ # rmmod module.ko
Djekmani@Bt3-security ~ # dmesg
```

وبكدا نكون عملنا chargement و dechargement لل Module تبعنا

Description Module

تستطيع وصف الموديل تبعك وتعملو حقوقه والاشياء دي وكل دا موجود في <linux/module.h>

نشوف المتال دا

```
#include <linux/module.h>
#include <linux/init.h>
MODULE_AUTHOR("djekmani4ever");
MODULE_DESCRIPTION("My first Module");
```

```
MODULE_SUPPORTED_DEVICE("ALL");
MODULE_LICENSE("Djekmani4ever's world license & GPL");

static

int __init mon_module_init(void)
{
printk(KERN_DEBUG "Hello djekmani !n");
return 0;
}

static

void __exit mon_module_cleanup(void)
{
printk(KERN_DEBUG "Goodbye djekmani !n");
}
module_init(mon_module_init);
module_exit(mon_module_cleanup);
```

كما لاحظ

```
MODULE_AUTHOR(nom) : اسم الكاتب
MODULE_DESCRIPTION(desc) : وصف للموديل
MODULE_SUPPORTED_DEVICE(dev) : العتاد الدي يدعمه الموديل
MODULE_LICENSE(type) : رخصة التابع لها الموديل
```

ممكن نستخرج المعلومات هدي عن طريق

```
modinfo Module_Name
```

Passage of parameters

الان فتحو شوي في الجزء الاخير دا عشان راج نبدا نشم فيه ريحت الجد .. من هنا يبدى الشغل ,لكي نتئكد من مرور بارامترات من الموديل تبعنا ما علينا الى ب function and macros يفيان بالغرض لاحظ معي :

```
module_param(nom, type, permissions)
MODULE_PARM_DESC(nom, desc)
```

هناك العديد من type مدعومة عشان ال parametre type

```
short (entier court, 2 octet)
int (entier, 4 octets)
long (entier long)
charp (chaînes de caractères)
```

لكي تتوضح الفكرة جيدا لاحظو معي المتال التالي :

```
#include <linux/module.h>
#include <linux/init.h>
MODULE_AUTHOR("djekmani4ever");
MODULE_DESCRIPTION("exemple Param");
MODULE_SUPPORTED_DEVICE("ALL");
MODULE_LICENSE("djekmani4ever's World");

static

int param;
module_param(param, int, 0);
MODULE_PARM_DESC(param, "My first Module");

static

int __init mon_module_init(void)
{
printk(KERN_DEBUG "Hello Djekmani !\n");
printk(KERN_DEBUG "param=%d !\n", param);
return 0;
}

static

void __exit mon_module_cleanup(void)
{
printk(KERN_DEBUG "Goodbye Djekmani !\n");
}
module_init(mon_module_init);
module_exit(mon_module_cleanup);
```

نجرب مرور البارامتر

```
Djekmani@Bt3-security ~ # insmod ./module.ko param=2
```

وهكدا نكون اكملنا مقدة بسيطة عن Kernel Modules ويمكن اعتبارها قاعدة عشان تدخل عالم الكرنل وخباياه وبرمجته .

بعد ان تعرفنا على بعض المداخل في برمجة الـModules هنا , الان سوف ندخل في بعض التفاصيل الاخرى لتكون تكملة لما سبق , والان سوف نتعرف على برمجة السواقات في حالة الـ caracter .

•المحتوى

1. Adding a driver to Kernel
2. Implementation of system calls
3. Open and release methods
4. Memory allocation

هدا فقط ما سوف نتطرق له في هدا الجزء من سلسلتنا .عندما نضيف driver الى النواة يجب على النظام ان يؤتر او يصدر رقم رئيسي , هدا الرقم لتحديد الـdriver . يتم تسجيل الـ Driver في النواة عندما يتحمل الـ Module يعني module loading , وهدا يتم على مستوى الكود داخل الدالة init_module() يتم المنادات داخل هده الدالة على دالة اخرى تعتبر نقطة بداية الـ dirver داخل الموديل register_chdev ونفس الامر عندما نريد ان نمسح dirver من النواة يكون الامر في نقطة نهاية الموديل cleanup_module يتم المنادات على دالة اخرى تعتبر كنقطة نهاية الـ dirver من الموديل وهي unregister_chdev . , هده الدوال معرفة داخل مكتبة <linux/fs.h> .

نمادج تطبيقية :

```
int register_chrdev(unsigned char major, const char *name, struct file_operations *fops);

int unregister_chrdev(unsigned int major, const char *name);
```

`register_chrdev`

- major : ل الرقم الرئيسيdirver , يمكن تجعله ديناميكي بجعله الرقم 0 يدل على دناميكيته
- name : اسم العتاد الموجود هنا

```
/proc/devices
```

- fops : هي عبارة عن pointer كـ تعرف الدوال المستخدمةsystem calls متل read ..open ..

`unregister_chrdev`

- major : ل الرقم الرئيسيdirver , يمكن تجعله ديناميكي بجعله الرقم 0 يدل على دناميكيته
- name : اسم العتاد الموجود هنا /proc/devices

```
struct file_operations fops =
{
.read = my_read_function,
.write = my_write_function,
.open = my_open_function,
.release = my_release_function
};
```

Implementation of system calls

```
static ssize_t my_read_function(struct file *file, char *buf, size_t count, loff_t *ppos)

{

printk(KERN_DEBUG “read()\n”);

return 0;

}

static ssize_t my_write_function(struct file *file, const char *buf, size_t count, loff_t *ppos)

{

printk(KERN_DEBUG “write()\n”);

return 0;

}
```

```
static int my_open_function(struct inode *inode, struct file *file)

{

printk(KERN_DEBUG “open()\n”);

return 0;

}

static int my_release_function(struct inode *inode, struct file *file)

{

printk(KERN_DEBUG “close()\n”);

return 0;

}
```

الـ stucture file معرفة في <linux/fs.h> وتمتل ملف مفتوح من طرف النواة عن طريق الـ syscall open() .

حقول مهمة داخل الكود الاخير :

* mode_t f_mode : يمتل حالة فتح الملف
* loff_t f_pos : المحل الدي يتم القراءة او الكتابة عليه
* unsigned int f_flags : flags files(O_NONBLOCK…)
* struct file_operations *f_op : العمليات المرتبطة بالملف

Open and release methods

بصفة عامة طريقة الـ open والـ release يفيدان في عدة عمليات منها :

- التحكم بالاخطاء على مستوى العتاد

- فتح واستخدام العتاد

- allocation و املاء strucuture الخاصة التي سوف تعوض في file->private_data

- استهلاك العتاد

- اطفاء العتاد (release)

```
#include <linux/module.h>
#include <linux/init.h>
#include <linux/fs.h>

MODULE_AUTHOR(”djekmani4ever”);
MODULE_DESCRIPTION(”my first driver”);
```

```
MODULE_SUPPORTED_DEVICE(”none”);
MODULE_LICENSE(”none”);

static int major = 254;

module_param(major, int, 0);
MODULE_PARM_DESC(major, “major number”);

static ssize_t my_read_function(struct file *file, char *buf, size_t count, loff_t *ppos)
{
printk(KERN_DEBUG “read()\n”);
return 0;
}

static ssize_t my_write_function(struct file *file, const char *buf, size_t count, loff_t *ppos)
{
printk(KERN_DEBUG “write()\n”);
return 0;
}

static int my_open_function(struct inode *inode, struct file *file)
{
printk(KERN_DEBUG “open()\n”);
return 0;
}

static int my_release_function(struct inode *inode, struct file *file)
{
printk(KERN_DEBUG “close()\n”);
return 0;
}

static struct file_operations fops =
{
read : my_read_function,
write : my_write_function,
open : my_open_function,
release : my_release_function /* correspond a close */
};

static int __init mon_module_init(void)
{
int ret;

ret = register_chrdev(major, “mydriver”, &fops);

if(ret < 0)
{
printk(KERN_WARNING “major error\n”);
```

```
return ret;
}

printk(KERN_DEBUG “mydriver load succes\n”);
return 0;
}

static void __exit mon_module_cleanup(void)
{
int ret;

ret = unregister_chrdev(major, “mydriver”);

if(ret < 0)
{
printk(KERN_WARNING “error unregister\n”);
}

printk(KERN_DEBUG “mydriver reload  succes\n”);
}

module_init(mon_module_init);
module_exit(mon_module_cleanup);
```

مباشرة بعد عملية ال compilation و chargement ,يجب تجربة رابطه على في وضع المستخدم , الان سوف نعمل ملف خاص

```
mknod /dev/mydriver.c 254 0
```

والان حان وقت تجربتنا الالى وان شاء الله لن تكون الاخيرة

```
cat mydriver.c > /dev/mydriver
```

بعدها ممكن التئكد بستعمال dmesg ومشاهدة الـ sys call زي open read release ..

ممكن ايضا كتابة كود صغير عمله يفتح العتاد ويرسل اليه بيانات تم يغلقه

```
#include <stdio.h>

#include <unistd.h>

#include <errno.h>

int main(void)

{

int file = open(”/dev/mydriver”, O_RDWR);

if(file < 0)
```

```
{
perror(”open”);
exit(errno);
}
write(file, “hello”, 6);
close(file);
return 0;
}
```

Memory allocation

في وضع النواة memory allocation . تستخدم بواسطة الدالة kmalloc و .ل desalocation تستعمل بواسطة kfree

```
arguments
GFP_KERNEL :لداكرة النواة normal allocation
GFP_USER :لحساب المستخدم memory allocation
#include <linux/slab.h>
buffer = kmalloc(64, GFP_KERNEL);
if(buffer == NULL)
{
printk(KERN_WARNING “problème kmalloc !\n”);
return -ENONEM;
}
kfree(buffer), buffer = NULL;
```

وبهدا اكون قد اتممة هدا الجزء ارجو ان اكون قد وفقت في شرح ما هو موجود بدهني لاني صراحة بجد صعوبة في شرح مواضيع برمجية ..ارجو ان يكون سلسا ومفهوما . وموعدنا في الجزء الاخير .

CurlFtpFS : FTP filesystem

Autor : Djekmani

CurlFtpFS : يتيح لك مشاهدة والدخول الى نظام ملفاتك عن بعد الدي كنت لا تستطيع الدخول اليه عن طريق بروتوكولات الاتصال العادية .(ssh . ftp .) هده اخدمة تعتمد اساسا على Curl و الـ FUSE

•مقدمة حول مشروع الـ FUSE

الهدف من مشروع الـ fuse هو تطوير برمجيات بلاعتماد على هدا المشروع التي تتيح لك مشاهده كل ملفات النظام وكل مدخلاته عن بعد .. ولتعرف على البرامج التي تستخدم نظام الـ fuse وتسمح ب mount لكل اشكال انظمة الملفات .

```
http://fuse.sourceforge.net/wiki/index.php/FileSystems
```

ونحن الان سوف ندرس برنامج من هده البرامج الدي هو Curlftpfs عن طريق نظام الـ fuse بستخدام بروتوكول الـ FTP .

•تتبيت

```
apt-get install curlftpfs
```

قد تأتي معها الحزم التالية المهم ممكن ان تتبتها ايضا **fuse-source fuse-utils libfuse2** مهمة جدا

لتشغيل هدا البرنامج لابد من شحن المودل دا في الداكرة

```
modprobe fuse
```

وبعدها الـ mount يشتغل بدون اي مشاكل عن طريق الـ commend التالية

```
curlftpfs ftp://127.0.0.1 /mnt -o user=user:pass
```

لكي تتفادا كتابة الكلمة السرية في سطر الاوامر لامور امنية ممكن ان تضيف هدا السطر في **netrc.**

```
machine ur-ftp-server login ur-user password ur-pass
```

ان لم تجد ملف /netrc. فعليك اضافته واكتب فيه السطر السابق . واعطيه تصريح لكي لا يستطيع مشاهدته سوى مستخدم واحد

```
chmod 600 ~\.netrc
```

بعدها ممكن الدخول لـ mount بكل سهولة

```
curlftpfs ftp://127.0.0.1 /mnt
```

وبهدا اكون قد انتهيت من شرح موجز صغير عن curlftpfs

ADUR[IT] TEAM MAGAZINE