Global Information Assurance Certification Paper

Data Charging Bypass
How your IDS can help

GIAC (GCIA) Gold Certification

Author: Hassan Mourad, Hassan.morad@gmail.com
Advisor: Chris Walker
Accepted: September 30th 2014

Abstract

The recent years witnessed an upsurge in the use of mobile Internet & data services, generating new revenue streams for mobile operators and increasing their annual revenue per user (ARPU). But with every opportunity comes risk. Fraudsters are always trying new techniques to bypass your charging controls and gain free access to such services, denying the operator of its revenue and damaging its reputation. In this paper we will examine some of those techniques, as well as how to use open source IDSs/IPSs to detect and possibly block these types of fraud.

1. Introduction

The recent increase in the number of smart devices, the introduction of high speed mobile connections (4G/LTE), as well as the hype in social networking has all led to the dramatic increase in mobile Internet traffic. In their visual networking index, Cisco states that, "Over half a billion (526 Million) mobile devices and connections were added in 2013 rising to a total of 7 billion" (Cisco, 2014, p 1).

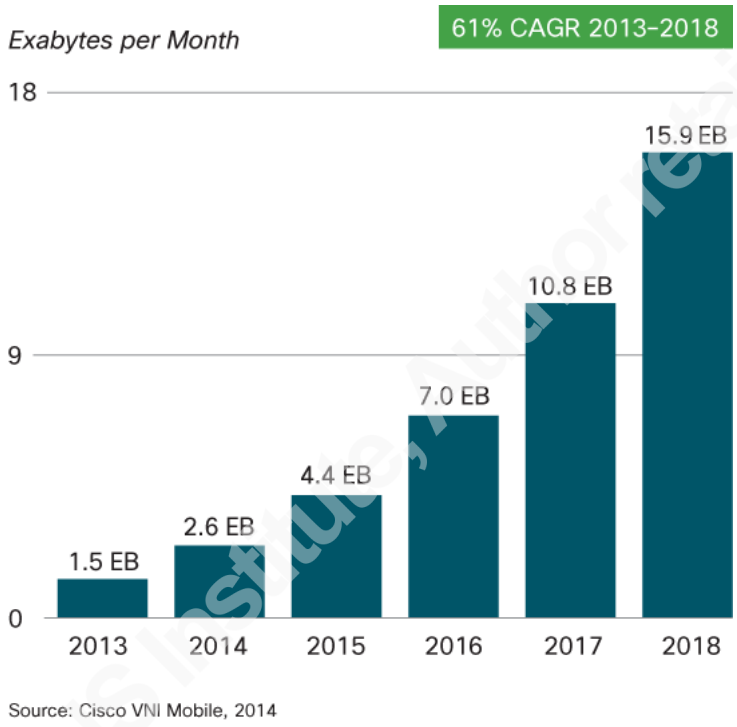


Figure 1. Cisco Forecasts 15.9 Exabytes per Month of Mobile Data Traffic by 2018

In the USA, according to Chetan Sharma consulting, "in Q4 2013, the mobile data revenues also eclipsed the 50% mark meaning that more than 50% of the mobile services revenues is now coming from data services" (Chetan Sharma, 2014). Clearly this increase presents a great opportunity to the Mobile Network Operators (MNOs) to increase their Annual Revenue Per User (ARPU) specially with the declining revenues of the traditional voice and SMS services.

Hassan Mourad, Hassan.morad@gmail.com

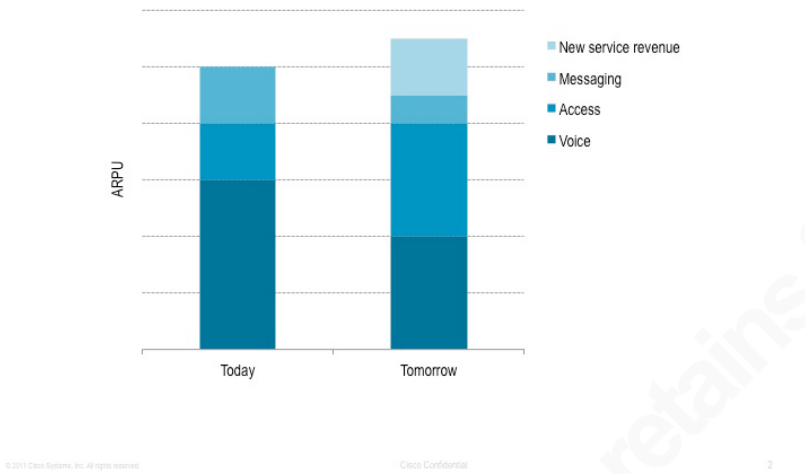


Figure 2. Arpu Evolution
Note: "Tomorrow" equates to roughly 2–5 years. Data is intended to show a trend.
Adapted from: Chetan Sharma Consulting, 2012

But with opportunity comes risk. Several attacks against the operator's charging process can jeopardize this revenue opportunity. It is in the Operator's absolute Interest to detect and prevent such attacks.

In this paper we will examine the typical attacks against the charging process and discuss how to use the available security tools to detect and possibly prevent such attacks

2. Data charging in a Mobile Operator

The operator's network is an exceptionally complex network. In the background, several network elements are contributing to the user's experience from the time he/she initiates a data connection, throughout his session, till he disconnects the session. It is beyond the scope of this paper to provide a comprehensive discussion of all network elements; however, we will briefly describe the main components that contribute to the charging process for user traffic.

One of the most important pieces of the charging process, and data access in general, is the Gateway GPRS Support node (GGSN). The GGSN acts as a gateway for the user traffic to access a Public Data Network (PDN), typically the Internet. When a subscriber connects to the Internet, a Packet Data Protocol (PDP) context is established with the GGSN. The PDP context is a data structure that contains subscriber's session Information, such as his IP address, International Mobile Subscriber Identity (IMSI), and his Mobile Station International Subscriber Directory Number (MSISDN), or simply his phone number.

A tunnel is established between the Serving GPRS Support Node (SGSN) and the GGSN. User traffic is encapsulated using GTP-U protocol. At the GGSN it is de-capsulated and sent to the PDN through the Gi interface (or the Gp in case of roaming). PDP establishment and termination occurs through the GTP-C protocol. (3GPP, 2014)

On 4G/LTE networks the functionality of the GGSN has been replaced by the Serving Gateway (SGW) and the PDN Gateway (PGW).

There are two types of charging, offline charging, or online charging. The 3GPP technical specification TS32.240 for charging principles defines offline charging as "a mechanism where charging information does not affect, in real time, the service rendered". In Online Charging Systems (OCS) on the other hand "the charging information can affect, in real time, the service rendered, therefore requiring direct interaction with bearer/session/service control is required". (3GPP, 2014, p. 18)

Another key component in the charging process is the Policy and Charging Enforcement Function (PCEF). The PCEF "enforces the policy and charging rules it

receives from the charging system and/or the Policy and Charging Rules Function" (PCRF) (3GPP, 2014). on the user traffic. For example it can throttle user traffic after his quota is exhausted, or blocks user traffic in case he has no sufficient balance. In real life the PCEF functionality exists on either the gateway (GGSN – PGW), or is provided by an external network element such as Deep Packet Inspection (DPI) solutions.

The 3GPP technical specification TS 23.002 identifies the Policy and Charging Rules Function (PCRF) as "a policy decision point for policy and charging control of service data flows/applications and IP bearer resources. The PCRF selects and provides the applicable policy and charging control decision to the PCEF and, if applicable, application detection and control decision to the TDF or PCEF with application detection and control feature". (3GPP, 2014)

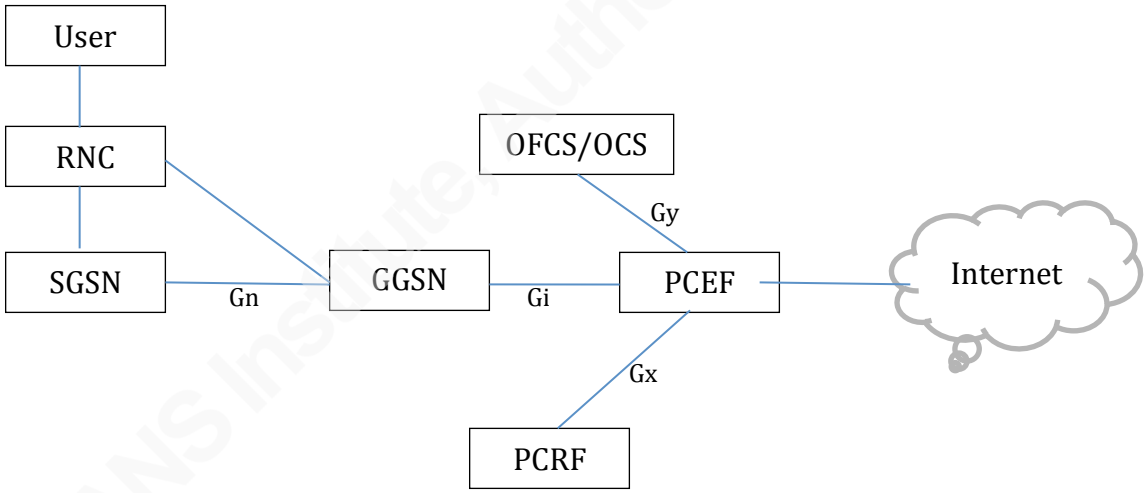


Figure 3: Typical 3G Network

3. Data charging attack techniques

Attacks against the charging process typically falls into two main categories, policy weakness attacks, and stress-testing attacks.

Policy weakness attacks targets misconfiguration or improper configuration of the charging policy. The golden rule in such attacks is that whatever you offer for free can be

used against you. Operators tend to offer free data services either because of commercial requirements or technical limitation. In this category we will discuss three different techniques used to bypass your charging; the free URL technique, traffic tunneling, and internal proxy misuse.

The second category relies on creating an exception condition in the charging system by overloading the system with specific traffic. In this category we will discuss issues resulting from overloading the system's accounting functionality.

For each attack, we will discuss the theory behind it, and the proper configuration to prevent it. We will then analyze the typical tools used to conduct this attack. Finally we will present how to use our security tools such as snort, OSSIM, and SilK to detect, and possibly prevent those attacks.

3.1. Free URL Bypass

What happens when a user is out of credit and the operator needs to inform him about that? One way they do this would be to redirect your web traffic to a captive portal to inform the user about his balance, and possibly allow him to add extra credit or purchase additional megabytes. But if the user is out of credit and have no megabytes left in your bucket, how can he access this portal?

Some operators offer their subscribers special buckets that allow them free access to a group of websites (e.g. free access to social networking, or web email, etc.). But how does the operator differentiate between those sites and the other sites that consume the subscriber's bucket.

Well, typically what the operator does is that they zero rate this URL. So for the first example requests destined for the operator's portal will be charged for free.

But what if an attacker was able to manipulate requests going to none free sites to appear as if they were going to free sites. If this happens the attacker will be able to obtain free unrestricted access to the Internet.

How can this be done? It all depends on how the charging rule is created. Ideally the charging rule needs to cross reference the free URL to the destination server IP. A proper rule for free access to a free URL should look like the following.

If URL = www.freehost.com & destination IP = free-host-ip then traffic = free.

But it is not always possible to build a charging rule like this, either because the charging system does not allow for complex rules or because the destination IP is dynamic. When this happens an attack surface is presented to the attacker to manipulate the requests into getting his free access.

3.1.1. The Attack

In normal http requests the URL is constructed by both the User Resource Identifier (URI) field and the host header. Below is what a normal GET request to www.example.com would look like.

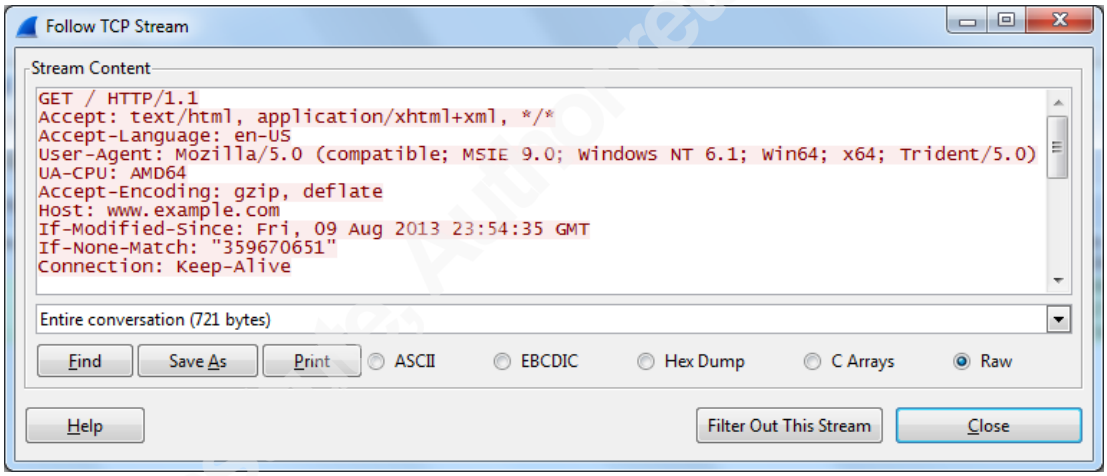


Figure 4. Normal HTTP get request

When using a proxy server to reach the same URL the request slightly changes. According to RFC2616 "an absolute URI is required when the request is being sent through a proxy" (IETF, 1999). The request to www.example.com would look like this

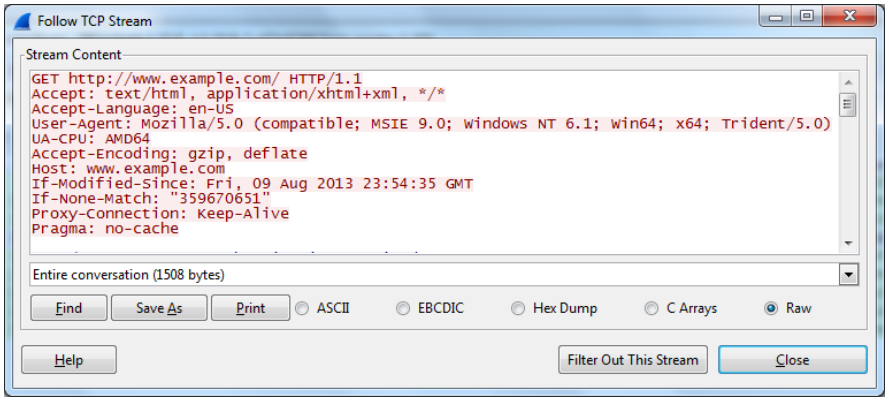


Figure 5. HTTP get request through proxy

An attacker can take this request and manipulate the original host field and replace it with the free URL host. The manipulated request would appear as follows

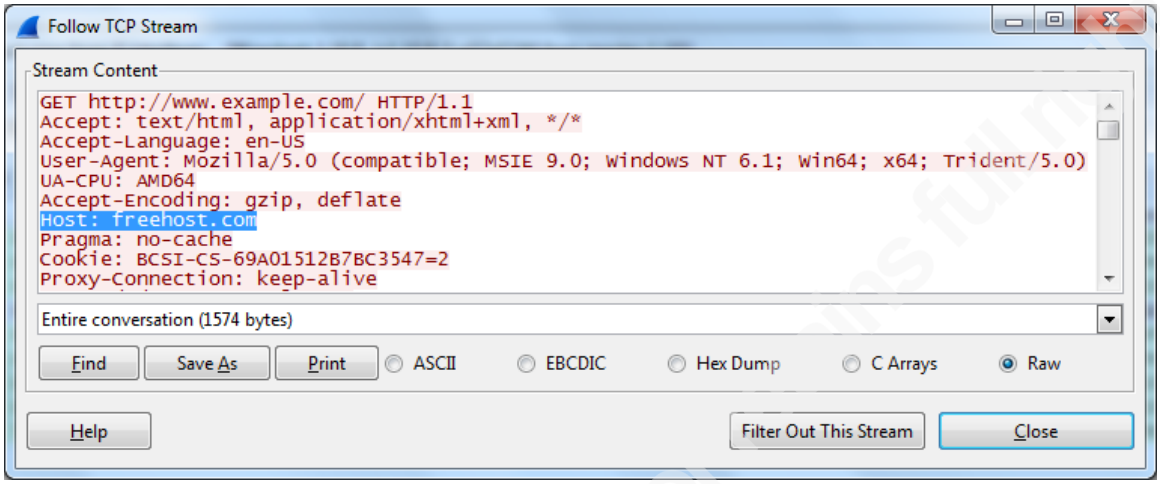


Figure 6. Manipulated HTTP get request adding free host

When this request is received by the proxy it is forwarded to the host server that is in the absolute URI field, and depending on the receiving server configuration, most servers would ignore the host header field and serve the request from the actual host in the URI field

Now looking at the last request from charging perspective, it might satisfy a charging rule that is only based on URL/Host, and hence this request will be charged for free although it is not going to the free destination

3.1.2. The Tools

Your Freedom

Your freedom is described by its authors as "The All-IN-One VPN-Tunneling, Firewall & Proxy bypassing, anonymization and anti-censorship solution". (Your-Freedom, n.d.)

Your freedom acts as a local proxy that the user points his browser and other programs to, it works by establishing a connection to one of your freedom servers. It supports VPN tunneling over multiple protocols, including HTTP, HTTPs, & DNS.

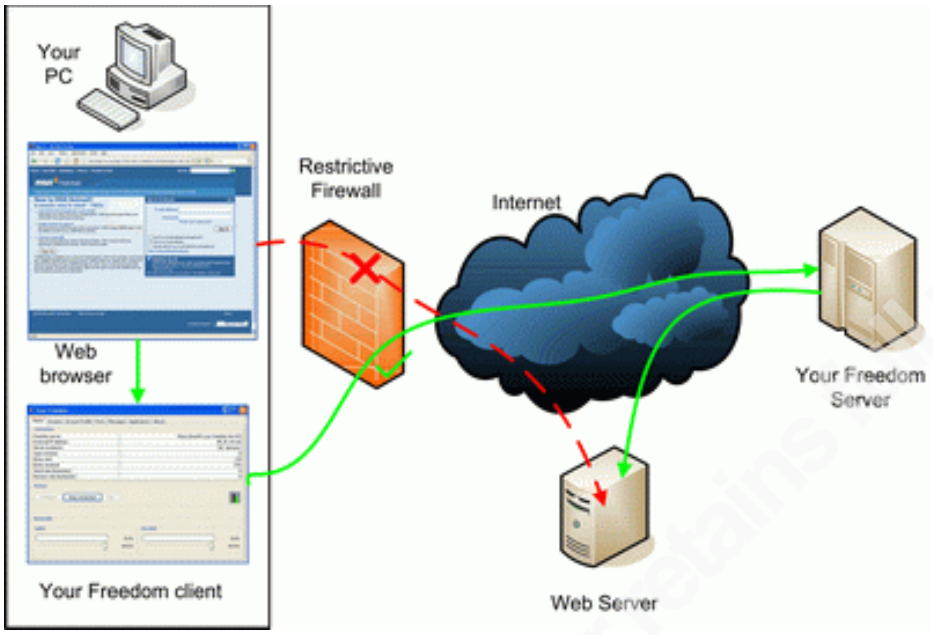


Figure 7. How Your-Freedom Works. Adapted from www.your-freenet.com

Quoting your freedom's website: "Will Your Freedom provide me free access to the Internet?

Probably yes, particularly if you use DNS tunneling mode. You may be able to get a working mobile phone connection when the provider does not want it to be working (no credit, special APNs, etc.), and you may be able to use wireless hotspots without paying for the usage. Your Freedom cannot differentiate between true censorship and merely commercial interests; if there is a way out, Your Freedom will find it and use it". ". (Your-Freedom, n.d.)

The DNS tunneling mode will be discussed in the next section, but for this specific attack what makes Your Freedom powerful is its ability to add free URLs/hosts to its configuration.

By adding the free URL to your freedom's server configuration, and pointing out the program to an external proxy, the program is in this way performing the Free URL fraud attack.

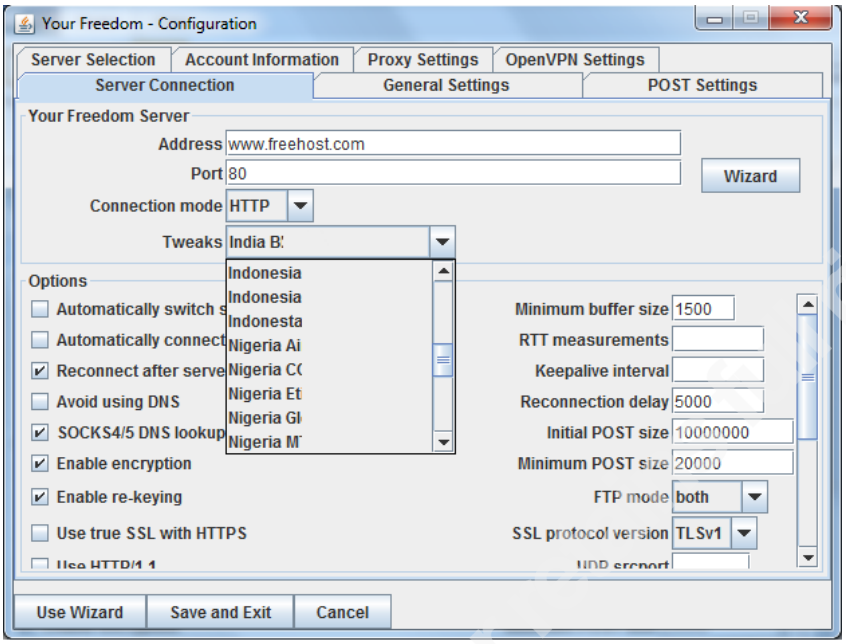


Figure 8. Your Freedom – Adding Free URL/ Tweeks

One of the best features regarding your Freedom is its "Tweaks", the program comes with a list of predefined tweaks or manipulations for some operators across the world that already attempts to give you free access to the Internet, by masquerade the packets as ones distend to free services

The only cavity about your freedom is that the free service it offers is limited in terms of speed and daily time usage. If you wish to remove this limitation you have to pay.

Proximetron

Proximetron is a free light weight local proxy that allows users to manipulate and filter http requests and responses based on different rules. The rules are located in a configuration file.

The configuration file uses simple regex to find and replace certain headers with the value of the user's choice. In the attack case the configuration file would match the host header and replace its original value with the free host. Below is an example of the configuration file.

```
[HTTP headers]
In = FALSE
Out = TRUE
Key = "URL: Un-Prefixer (Out)"
Match = "[^\]+\w[^a-z]((http|ftp)(%3A|:)(%2F|/)[^&]+)\1"
Replace = "$JUMP($UESC(\1))"

In = FALSE
Out = TRUE
Key = "URL: Enable Keyword search (Out)"
URL = "[^./]+/(^?)&\w[a-z]&$JUMP(http://www.google.com/search?q=\h)"

In = TRUE
Out = FALSE
Key = "Content-Type: Fix MIME types (In)"
Match = "text/*&$URL(https+://[^/]+*.([a-z0-9]+{2,5}(^?)&&$LST(MIME-List)))"
Replace = "\0"

In = FALSE
Out = TRUE
Key = "Forwarded: (out)"
Replace = "\h"

In = TRUE
Out = TRUE
Key = "Host"
Replace = "freehost.com"

In = FALSE
Out = TRUE
Key = "X-Forwarded-For (out):"
Replace = "yahoo.com, microsoft.com, netscape.com, aol.com, \h"


[Patterns]
Name = "Anti-Auto-Refresher"
Active = TRUE
```

Figure 9. Proximetron Configuration file

3.1.3. The Defense

Well, this is one of the easiest attacks to detect; you are simply looking for web requests going to a free URL, while the packet itself is not going to the right destination. Remember, this should have been done in the charging rule but it was not.

Here is a sample snort rule to detect such an attack

alert tcp any any -> !$Free_Destination 80 (msg:"Free URL Fraud using freehost.com"; content:"freehost.com"; nocase; sid:900000; rev:1;)

The rule is checking for the content freehost.com in an http packet that is not going to the IP addresses of free destinations. Of course you will have to define the Free_Destination variable to include the IP addresses of the free servers

3.1.4. Additional consideration

URL encoding problem

One thing you need to consider with http requests is how the charging system handles character encoding and transformation. Does it work by doing an exact match for

the URL characters, or does it also understand encoded characters. In the case of the URL using an IP address, does it only match the dotted decimal representation and treat it as free, or will it also accept the decimal or octal representation of the IP.

For example if an operator is not charging the URLs for the address 10.10.10.10, what happens if an attacker sends the traffic to the URL http://0x0a.0x0a.0x0a.0x0a (Hex representation for the 10.10.10.10), or the URL http://168430090 (decimal representation for 10.10.10.10). Will the charging system treat those as free or not? If it does, the operator needs to also add rules on snort for those transformations

Your freedom for example as part of its tweaks will try to transform a given IP to different forms to get you access. Below is a screenshot of a request using a transformation of the original IP 10.58.168.4

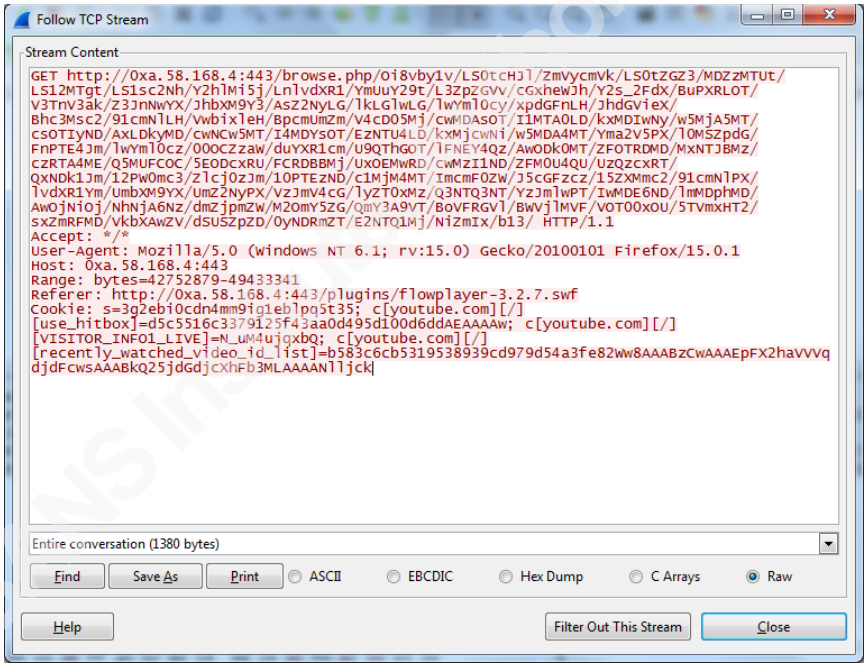


Figure 10. Hex transformation for a Free IP

In case the charging system treats other forms of encoding and transformations as free, you will need to add rules for those as well in your IDS.

Hassan Mourad, Hassan.morad@gmail.com

Modding mobile programs for the attack

So far what we have examined programs to be used on PCs to gain free internet access, and although no similar programs exists for this attack on the mobile side, there is nothing preventing someone from developing this at any point of time.

However this is not the only way you can gain free access using your mobile. Some programs can be modified to add free URLs to its requests. Opera mini for example is a program that has been famous of being modified to allow this to happen.

We will examine at a later section how to analyze a modded operamini program to find the free URL that it uses.

3.2. Tunneling problem (DNS/ICMP)

Tunneling traffic through different protocols, in particular, through DNS or ICMP, has always been a favorite way for attackers to bypass firewalls and censorship. In a hotspot setup one thing hotspot owners occasionally did was to filter/redirect web traffic to the captive portal, but allow DNS or ICMP outside of the hotspot. This allowed an attacker to tunnel traffic using those protocols and gain Internet access.

DNS packets are typically (not necessarily) small in volume, however they tend to be high on the number of transactions, a factor affecting the sizing of some charging systems. Some operators as a trade-off accepts not charging those packets in favor of the system being able to handle bigger volumes with fewer transactions.

This can be done by excluding the operators own DNSs from charging, excluding all DNS traffic from charging (Traffic going to port 53 UDP), or just routing all DNS traffic outside the charging cloud.

This setup however opens the venue for attackers to use it to gain uncharged access to the internet using tunneling techniques.

3.2.1. The Tools

Your freedom again

Your freedom will attempt to connect to some of its VPN servers over UDP port 53, if this traffic is not charged the connection attempt will be successful

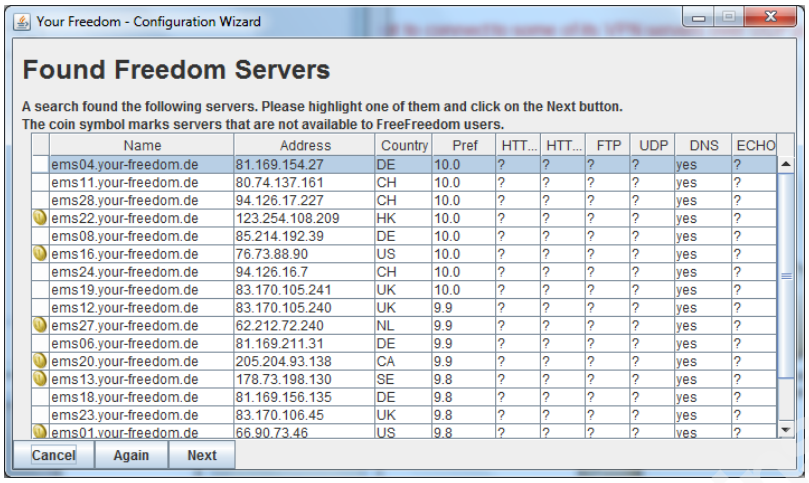


Figure 11. Your Freedom DNS servers

VPN over DNS

VPN over DNS is a program that allows you to tunnel your traffic over DNS. The nice thing about it is that beside its Windows, & MAC OS versions, it also has an android version that you can use on your phone

Wi-Free

Wi-free allows the user to tunnel traffic over DNS (Port 53 UDP) and ICMP. It however requires you to purchase an account to be able to use this service

3.2.2. The Defense

System for Internet-level knowledge (SiLK) is an open-source suite of tools that allows for efficient collection, storage, and analysis of netflow information. "It converts the data into a more space efficient format, recording the packed records into service-specific binary flat files.

The analysis suite consists of tools which read these flat files and perform various query operations, ranging from per-record filtering to statistical analysis of groups of records. The analysis tools interoperate using pipes, allowing a user to develop a relatively sophisticated query from a simple beginning". (CERT NetSA, n. d.)

You need to configure your routers to send flow records to SiLK. A flow record includes source/destination IPs, source/destination ports, protocol, TCP flags, total bytes and packets, start time, end time, duration, and acquiring sensor identification.

The main tool in the Silk suite is the rwfilter. Rwfilter takes reads the binary files that contains the flow information, applies specific filters to the flow records, and provides the output to the user. The output can be piped to other SiLK tools such as rwstats to obtain statistics about the traffic.

Rwstats summarizes SiLK flow records and provides statistical information about the flows such as top sources or top destinations (CERT NetSA, n. d.).They can be used to focus on source IPs with huge number of DNS flows.

You can use SiLK to detect all users that have high DNS (Port 53 UDP) traffic going to the Internet. By applying simple SiLK filters you can identify users that have anomalous traffic pattern and crosscheck their usage with their spending to identify fraudsters.

Below is a sample silk rule that will display the top 100 source addresses with DNS flows (UDP port 53)

rwfilter flowfile.silk --proto=17 --dport=53 --pass=stdout | rwstats --fields saddress --count=100 --flows

And here is one for ICMP

rwfilter flowfile.silk --proto=1 --pass=stdout | rwstats --fields saddress --count=100 --flows

Using snort to detect DNS/ICMP tunneling

Snort already comes with a set of rules that can be used to detect DNS tunneling. It also has some rules that are specific to some of the programs that are used in tunneling

Here is a list of snort rules that can be used to identify traffic tunneling. Note however that traffic tunneling can be a normal behavior for legitimate users. You must crosscheck this against the actual user's charging to find if it is being used for fraud or not.

(1:21853) APP-DETECT ptunnel icmp proxy

(1:27046) APP-DETECT iodine dns tunnelling handshake server ACK

(1:27700) APP-DETECT NSTX DNS tunnel outbound connection attempt

Hassan Mourad, Hassan.morad@gmail.com

(1:19471) POLICY-OTHER dnstunnel v0.5 outbound traffic detected

(1:25983) INDICATOR-OBFUSCATION DNS tunneling attempt

3.2.3. Additional consideration

Traffic tunneling using Operator's DNS

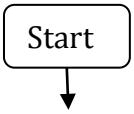
To avoid the previous scenario, some operators would charge traffic going to DNS servers that they do not control, but they may not charge DNS traffic going to their own DNS infrastructure under the false assumption that their controlled DNS servers cannot be used to tunnel traffic. Unfortunately, it is still possible to use legitimate DNS servers for traffic tunneling.

Before we explain how this is done let us look at a typical name resolution example.

First, when a user attempts a connection to domainX.com, he will send a DNS request to the DNS server querying for the IP address of domainX.com. The DNS server first checks its local cache for the IP of domain.com from previous translations. If it finds it, it will directly send the reply to the user.

If the DNS server does not have the mapping in its cache, it will then query the root DNS servers for the authoritative name server responsible for domainX.com. The root server would return the NS record for this domain. The DNS server will then send a query for the name resolution to the authoritative name server. The authoritative name server will reply with domainx.com's IP, and the operator's DNS server will send the query response to the user.

In a way, a user request has been delivered to a server that is not controlled by the operator, and the response form this server has been delivered to the user.



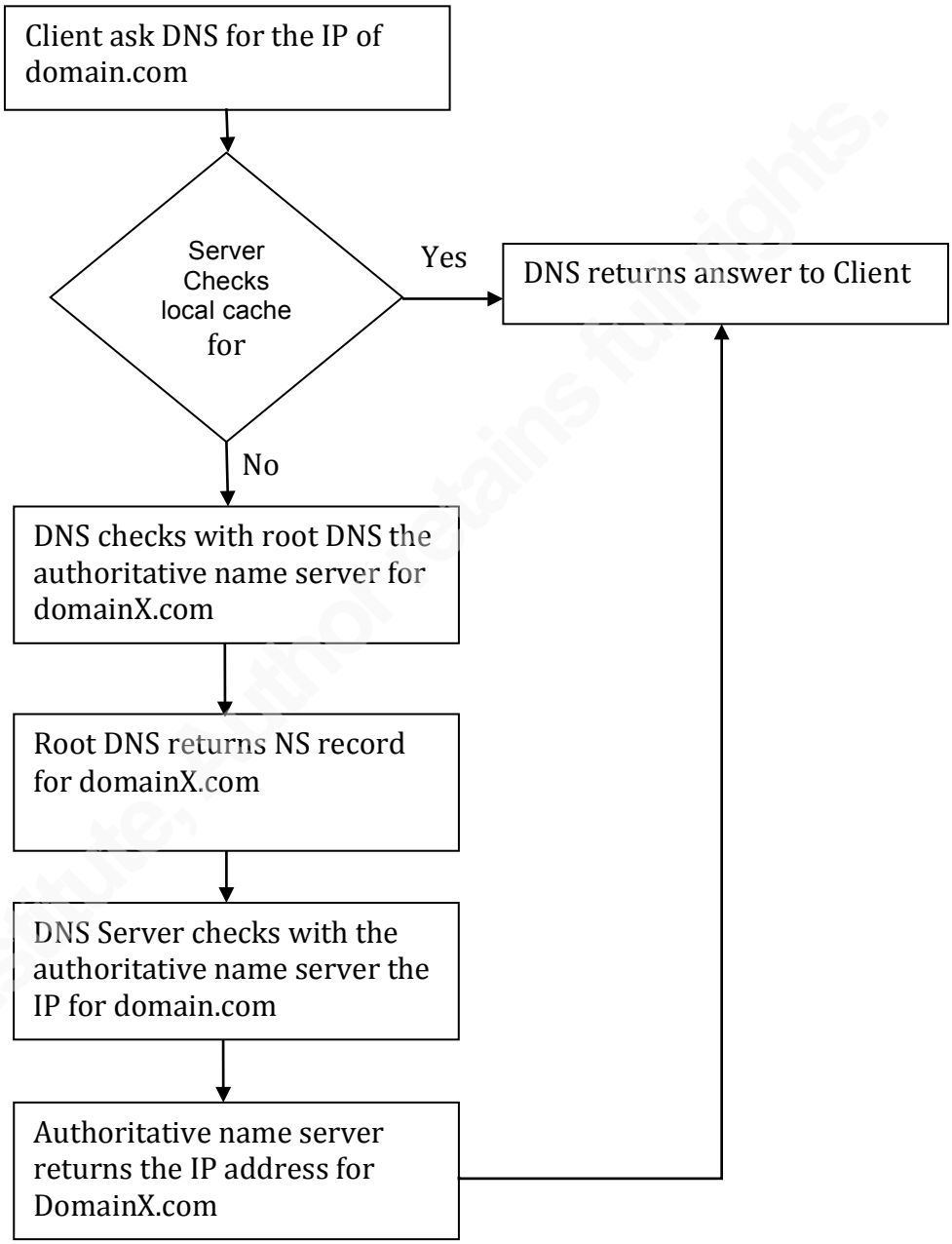


Figure 12. Simplified DNS resolution process

Now here is how the attack works. First the attacker needs to own a domain name, something he can buy for as low as 2 USD from any public registrar. Then he needs to have an authoritative name server for this domain under his control, which he can host for free using amazon web services for example.

Hassan Mourad, Hassan.morad@gmail.com

The attacker is trying to access www.example.com for free. What he can do is to encode this request using for example. He can then request DNS record for {base64}.domainx.com. The DNS will check the authoritative DNS server for domainx.com which is controlled by the attacker, and will forward the request to it. The attacker's server will receive the resolution request and decode the base64 encoded request. It will then send the request on behalf of the user to www.example.com, encode the response, and send it as a response to the user query.

The TTL value for the response is typically set to a low value to prevent the operator's DNS server from caching the response. When the query response is received by the attacker it decodes the response and forwards it to the user's browser. In this attack the query is done using record types that allows for the bigger response such as null, private and TXT records. If the base64 value of the request is longer than the allowed length for the name field, the value is sent over multiple requests

Iodine is a tool that implements this attack. The attacker installs the client version of it that establishes a tunnel to the server version of iodine that is installed on the attacker controlled authoritative name server.

3.2.3.1. The Defense

Again SiLK can be used to detect high volume of DNS traffic going to your own DNS servers. The SiLK rule will look as follows

rwfilter flowfile.silk --proto=17 --dport=53 –daddress=ip_of_you_dns --pass=stdout | rwstats --fields saddress --count=100 --flows

Another option is to use snort to look for high number of DNS Null, Private, or TXT requests, combining the snort rule with thresholding. The following rule for example checks for source addresses with more than 20 TXT DNS requests within 60 seconds.

alert udp any any -> $DNS_Servers 53 (msg:"High TXT requests - Potential DNS Tunneling"; content:"|01 00|"; offset:2; within :4; content:"|00 00 10 00 01|"; offset:12; within:255; threshold: type threshold, track by_src, count 20, seconds 60;)

3.3. Internal Proxy Misuse

As mentioned previously, the operator's network is a very complex network, and typically offers multiple internal services to the customers. It usually has multiple portals offering different services to its customers (e.g. bill check, quota usage, free downloads, etc.)

But what happens if one of those services/servers is misconfigured to allow proxying of traffic and those servers have their own (of course uncharged) Internet access.

Attackers will scan the operator's internal network trying to identify one of those misconfigured servers, or they would try to hack those services and install a proxy service on them. When this happens you end up with an open uncharged channel to the Internet.

3.3.1. The Defense

Snort has multiple rules that can detect proxying activities. Below is a list of the current snort rules that can detect proxy usage.

(1:19472) POLICY-OTHER proxytunnel proxy connection detected

(1:19473) POLICY-OTHER stunnel proxy connection detected

(1:19475) POLICY-OTHER proxycgi proxy connection detected

(1:20136) POLICY-OTHER Glype proxy usage detected

(119:17) HI_CLIENT_PROXY_USE

But, proxy usage is normal for Internet users and is used in multiple legitimate cases. Enabling those filters in their default state on all traffic would simply overwhelm the operators with a huge amount of alerts.

But we are not looking for any proxy activity. We are looking for proxy traffic that is targeting internal network. By simply modifying the rules to trigger on proxy traffic going to the $Homenet, we can easily detect such attack. Of course you need to insure that the $homenet variable includes all servers that you offer to your customers.

3.3.2. Additional consideration

WAP gateway as a proxy (MMS problem)

One of the systems used in an operator network is the WAP gateway. The WAP gateway is used to support legacy handsets that use the WAP protocol for Internet access. It acts as a proxy for those users to access the Internet. The users typically connects to the WAP gateway using a different access point name (APN) that is preconfigured to use the WAP address as its proxy.

Another common use of the WAP gateway is in Multimedia message (MMS) handling. The WAP gateway will act as a proxy (relay) to deliver user MMSs to the Multimedia Messaging Center (MMSC). The MMS traffic is usually sent using a different APN that has the WAP gateway configured as a proxy, and has the URL of the MMSC in it.

Since an MMS is typically limited in its size, some operators tend to charge it via count, ignoring the actual volume of the MMS.

But what happens if an attacker uses the MMS APN, the WAP gateway as a proxy, but requests traffic from the Internet. The WAP gateway has internet access to support WAP users, and unless it implements a check to verify that traffic coming from an MMS APN is actually destined to the MMSC, it will pass the traffic to the Internet. With the MMS APN traffic volume set to free, we get a fraud condition.

3.3.2.1. The Defense

Operator needs to look for traffic using MMS APN that does not contain the MMSC URL to detect this type of fraud. This can simply be done by snort using the negate operator "!". A rule to detect such attack would look like this

Alert $MMSAPN any > $WAP_Gateway any (msg: "MMS Fraud Attempt"; content: !"http://MMS_URL"; nocase;)

Another possibility to detect this is by using a feature named header enrichment, often found in the operator's network to add some headers such as user's MSISDN & APN name to HTTP packets for different purposes.

By checking on the Internet gateway for packets that has the MMS APN header you can detect traffic attackers that successfully used the MMS APN to get Internet access.

One cavity in this approach is when the WAP gateway is used to connect to SSL websites. The WAP gateway would use the HTTP connect method to connect to SSL websites, hence the traffic would be encrypted after the WAP gateway, and the APN header will not be visible to your IDS. To address this cavity you can create a rule that checks for connect method from the MMS APN to the WAP gateway. The rule would look like this

Alert $MMS_APN any > $WAP_Gateway any (msg: "Connect attempt using MMS APN"; content: "connect"; nocase;)

3.4. Accounting problems

In order to be able to charge you, a unique identifier per subscriber needs to be used. In a mobile operator's environment this unique identifier is the MSISDN.

When you make a data connection you are assigned an IP address from a pool of IP addresses. This assignment is typically dynamic. So how would the charging system tie the traffic from an always changing IP address to the MSISDN?

The charging setup, specially the one with external PCEF component relies heavily on receiving the IP to MSISDN mapping from the GGSN. When the user connects to the network an accounting start message is sent to the charging system, telling it that this specific MSISDN has been assigned this IP address. The charging system would then keep this mapping in its database, and will use it to charge traffic seen from this IP address to the MSISDN that matches it. When the user disconnects from the operator network a release message is sent to the charging system telling it that the IP address is no longer assigned to this MSISDN. The IP address can then be used with other users.

As such, charging process relies heavily on those accounting messages to properly charge the user. So what happens if the charging setup cannot coup with the number of messages that it receives. For example, what happens if a certain user disconnects from

the network and the charging system did not process the release message for this IP, and another user is assigned the same IP address? An exception is created and depending on the system configuration the first user might get charged for the traffic of the second user.

Another example is, if a user connects and his assignment message is not received or not processed by the charging system, again depending on the system configuration the user might get open access to the Internet.

3.4.1. The Attack

The concept behind the attack is to try to induce an exceptional condition for how accounting messages are handled by the charging setup. An attacker can keep connecting and disconnecting from the operator's network, essentially sending a high rate of assignment and release messages to the charging system. When this is done at the peak period, and with a high number of attackers trying this simultaneously, there is a chance that the attacker will cause the needed exception to get his free connection to the Internet.

Below is a simple batch script that can be used to automate the task of disconnecting and connecting to the network

```
:connect
rasdial operatorconnection
:chk
ping -n 3 -w 300 41.128.200.55
ping -n 1 -w 300 41.128.200.55 | find /c "Reply from 41.128.200.55" > alert.txt
if not ERRORLEVEL=1 (goto end:)
rasdial operatorconnection /disconnect
goto connect:
:end
Echo "Success"
ping -t 41.128.200.55
```

The script first connects to the operator's network using the command rasdial. It then checks if it can get traffic through by pinging any IP address over the Internet that is typically reachable. Since the attacker has no credit, under normal situation he should not get a reply from the external server, but if that happens and a reply is received this would essentially mean that an exception has occurred to the charging system, and that he now has an open session to the Internet.

If a reply is not received, the script disconnects from the network using the command rasdial /disconnect. It then loops and starts over with the connect-check-disconnect process.

3.4.2. The Defense

To reduce the attack opportunity, the operators can enable accounting interim messages. This sends accounting messages that included the IP-MSISDN mapping periodically to the charging system. The charging system can use these messages to update its mapping database, effectively correcting any exception that occurred and denying the attacker form his open session.

But interim messages comes with the cost of overloading the sending and receiving systems, and depending on system sizing this can be a cost that the operator cannot afford.

When this is the case, the operator is reduced to trying to detect such attacks and handle the fraudulent users through a different process, may be black list those users, or report them to the proper authorities. This is where OSSIM comes into play

What is OSSIM

OSSIM or Open Source SIEM, is an open source security incident and event management offered by alien vault, a company known for its security products in the area of security management and threat intelligence.

For legal and regulatory purposes, the mobile operators keep a log of when an MSISDN is assigned or releases an IP address. By forwarding those logs to OSSIM, the operator can create a simple rule to detect the attack behavior of the Accounting attacks.

Depending on the source of the log you may need to create a custom plugin for OSSIM to properly parse the log file.

Assuming the log entry looks like the following

Oct 31 08:08:08 User=01222444871 IP=10.10.10.10 Msg=Assigned

Oct 31 08:09:08 User=01222444871 IP=10.10.10.10 Msg=Released

The plugin file will contain the following

Regexp=(?P<date>\w{3}\s\d{1,2}\s\d\d:\d\d:\d\d)\sUser\=(?P<username>\d{11})\s\IP=(?P<src_ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})\sMsg=(?P<msg>\S+)

date={normalize_date($date)}

plugin_sid=1

src_ip={resolv($src_ip)}

username={$username}

userdata1={$msg}

You will need to activate the plugin from OSSIM main configuration file under /etc/ossim/agent/config.cfg by adding the following line in the plugins section

Accounting=/etc/ossim/agent/plugins/accounting.cfg

All the operator needs to do is to check for an MSISDN that has multiple assignment messages in a short time period. For example using OSSIM you can check if an MSISDN has 10 Assignment log messages in 5 minutes period. You can do this in OSSIM by building a custom directive to detect this.

Hassan Mourad, Hassan.morad@gmail.com

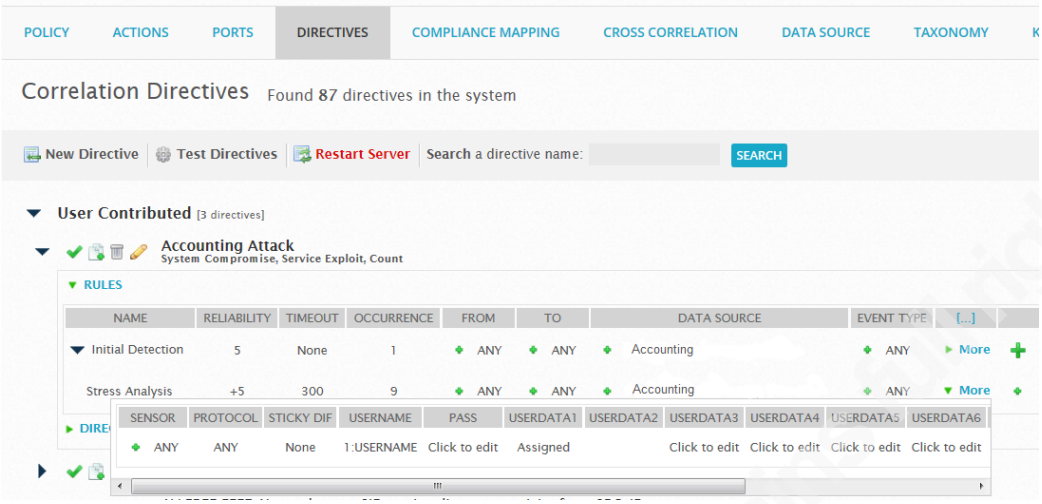


Figure 12. OSSIM Directive

Although the behavior can happen legitimately under very bad network conditions (e.g. strong radio interference, or improper handovers), this is something that the operator can easily identify, but in most of the cases this simple rule will yield all MSISDNs of the attackers attempting this attack.

4. Using Internet to find your exposure

The internet knows everything. If you are an operator and want to know if any of such attacks is possible on your network, you should simply ask the Internet.

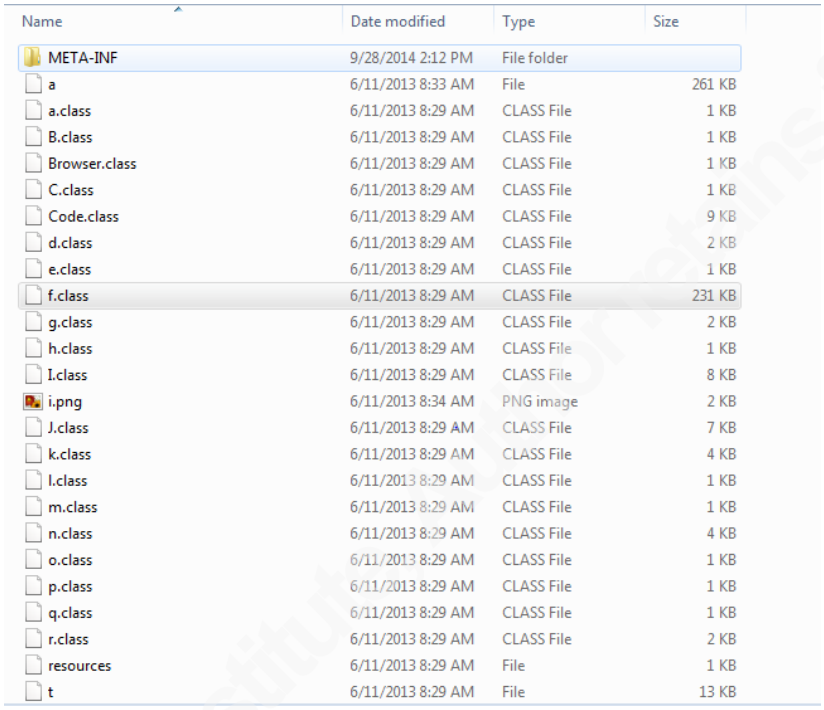
Try searching for your operator's name and the key word "free Internet" and you could be surprised of how many results you can find that is not part of your commercial offerings. The bad guys are always good in sharing, and there are plenty of forums on the Internet that discuss the techniques and the tools that can be used to gain free access to your network. You should also try this search in your own language.

4.1. Analysis of fraud programs

In some case, you will find the fraud forums offering programs that have been modified to gain free access to your internet possibly using a free URL, and you need a quick way to analyze he program to know the URL it is using. These programs are typically modified to add your free URL in its requests. You simply need to know where

to look for to get this free URL. Below is an example of how to analyze an opera mini jar file that has been modified to use a free URL.

First, you would either need to decompile the file to get the actual executed code, or another simple method would be to rename the jar file to zip and extract the various files used by this program. Here is the typical file structure that you will see.



Name	Date modified	Type	Size
META-INF	9/28/2014 2:12 PM	File folder	
a	6/11/2013 8:33 AM	File	261 KB
a.class	6/11/2013 8:29 AM	CLASS File	1 KB
B.class	6/11/2013 8:29 AM	CLASS File	1 KB
Browser.class	6/11/2013 8:29 AM	CLASS File	1 KB
C.class	6/11/2013 8:29 AM	CLASS File	1 KB
Code.class	6/11/2013 8:29 AM	CLASS File	9 KB
d.class	6/11/2013 8:29 AM	CLASS File	2 KB
e.class	6/11/2013 8:29 AM	CLASS File	1 KB
f.class	6/11/2013 8:29 AM	CLASS File	231 KB
g.class	6/11/2013 8:29 AM	CLASS File	2 KB
h.class	6/11/2013 8:29 AM	CLASS File	1 KB
I.class	6/11/2013 8:29 AM	CLASS File	8 KB
i.png	6/11/2013 8:34 AM	PNG image	2 KB
J.class	6/11/2013 8:29 AM	CLASS File	7 KB
k.class	6/11/2013 8:29 AM	CLASS File	4 KB
l.class	6/11/2013 8:29 AM	CLASS File	1 KB
m.class	6/11/2013 8:29 AM	CLASS File	1 KB
n.class	6/11/2013 8:29 AM	CLASS File	4 KB
o.class	6/11/2013 8:29 AM	CLASS File	1 KB
p.class	6/11/2013 8:29 AM	CLASS File	1 KB
q.class	6/11/2013 8:29 AM	CLASS File	1 KB
r.class	6/11/2013 8:29 AM	CLASS File	2 KB
resources	6/11/2013 8:29 AM	File	1 KB
t	6/11/2013 8:29 AM	File	13 KB

Figure 13. Extracting a modified Jar file for analysis

You then need to examine each individual file to get the used URL or host. You can simply do this by opening the file in a text or hex editor and searching for the words "http://" or "host". Focus on the big files first. In most of the cases this will yield something that looks like "http://free_url". This way, you can identify the free URL, now you can correct your charging rules, or add a snort rule as the ones discussed in section 3.1

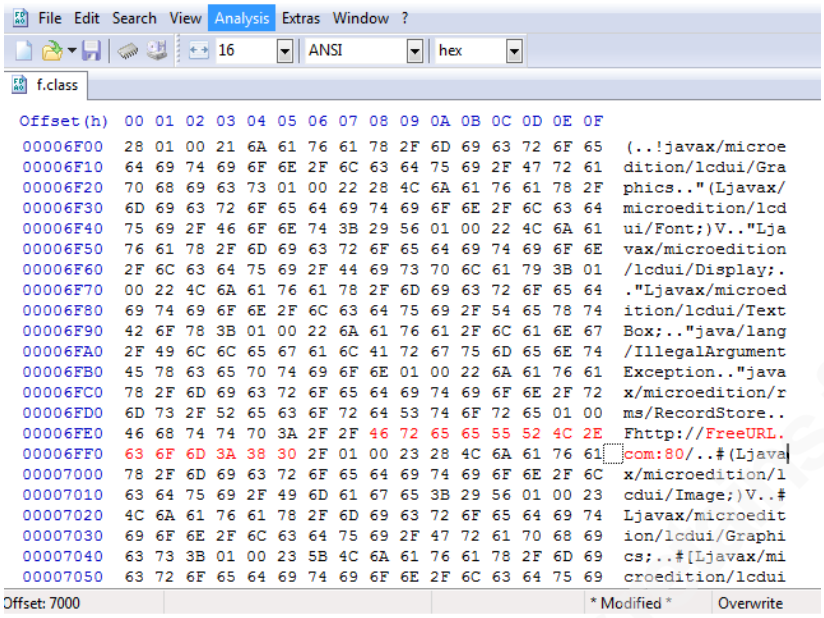


Figure 14. Searching a modified file for Free server

In some cases the programs will be packed using different packers. In this case directly searching for the text will fail. You need first to unpack the executable before trying your searches. If the URL is obfuscated, your only option would be to decompile the code to reverse the obfuscation code.

5. Conclusion

Data services are one of the most promising revenue streams to the operators. Operators need to strive to protect their revenue stream and insure that it is properly charged. Attackers will always try to find ways to beat your defenses and they will always find the one area you missed. You need to master your controls and know where to look for if you want to win this everlasting war. Tools like snort, OSSIM, SiLK and the Internet are among a lot of tools that you can add to your arsenal to tilt the balance to your side.

Hassan Mourad, Hassan.morad@gmail.com

References

Cisco (2014, February). Cisco Visual Networking Index: Global Mobile Data Forecast Update, 2013 – 2018, 1. Retrieved from: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.pdf

Chetan Sharma (2014). US Mobile Data Market Update – Q4 2013. Retrieved from http://www.chetansharma.com/US_Wireless_Market_Q4_2013_Update_March_2014_Chetan_Sharma_Consulting.pdf

3rd Generation Partnership project (2014, September 22), TS 23.060 General Packet Radio Service (GPRS); Service description; Stage 2, Release 12. Retrieved from: http://www.3gpp.org/ftp/Specs/archive/23_series/23.060/23060-c60.zip

3rd Generation Partnership project (2014, June 24), TS 23.002 Network Architecture, Release 12. Retrieved from: http://www.3gpp.org/ftp/Specs/archive/23_series/23.002/23002-c50.zip

3rd Generation Partnership project (2014, September 26), TS 32.240 Charging Architecture and Principles, Release 12, p18. Retrieved from http://www.3gpp.org/ftp/Specs/archive/32_series/32.240/32240-c50.zip

IETF (1999). RFC 2616, Hypertext Transfer Protocol -- HTTP/1.1, p35. Retrieved from: https://www.ietf.org/rfc/rfc2616.txt

Your-Freenet (n. d.), Retrieved on 2014, September from http://www.your-freenet.com

CERT Network Situational Awareness Team, CERT NetSA (n. d.), Retrieved on 2014 September from https://tools.netsa.cert.org/silk/index.html

CERT Network Situational Awareness Team, CERT NetSA (n. d.), Retrieved on 2014 September from https://tools.netsa.cert.org/silk/docs.html

3rd Generation Partnership Project (2002, June), TS 23.140 Multimedia Messaging Service (MMS); Functional description; Stage 2. Retrieved on 2014 September from http://www.3gpp.org/ftp/Specs/archive/23_series/23.140/23140-310.zip

Appendix A

Glossary of Terms

- APN: Access Point Name
- ARPU: Annual Revenue Per User
- CAGR: Compound Annual Growth Rate
- DPI: Deep Packet Inspection
- GGSN: Gateway GPRS Support Node
- MSISDN: Mobile Station International Subscriber Directory Name
- OCS: Online Charging System
- OFCS: Offline Charging System
- PCEF: Policy and Charging Enforcement Function
- PCRF: Policy and Charging Rules Function
- P-GW: Packet Data Network Gateway
- RNC: Radio Network Controller
- SGSN: Serving GPRS Support Node
- S-GW: Serving Gateway (4G)
- TDF: Traffic Detection Function