

NET SQUARE MACHINE TEST

- 1.encrypt/hash/encode 17/A3
- a. Decode as base 62 (4times) + Hex

Scopulus

Your Business Knowledge

TaxesArticlesToolsJargonContact

Tax Rates

Business Tools

Business Articles

Business Jargon

Free Updates

About us

Search

Contact us

Accessibility

Privacy/Cookie Policy

Legal Notice

Base64 Encoder Decoder

Simple Tools used by students and programmers Worldwide.

Use this encode/decode online tool to encode and decode a string, binary, decimal, hexadecimal, words, password or phrases, and a few more useful for programmers and web de
entities (special characters) name codes and number codes. See below for list and instructions. More Coming Soon! or you can contact us and make a suggestion. Back to busi

Base64 - encodes/ decode a string to MIME base64 format.

Encoder/Decoder

Select from 21

Base64

Text Encode/Decode:

4e65742d536563757265233234383930343030313023

EncodeDecode

Scopulus

Your Business Knowledge

TaxesArticlesToolsJargonContact

Tax Rates

Business Tools

Business Articles

Business Jargon

Free Updates

About us

Search

Contact us

Accessibility

Privacy/Cookie Policy

Legal Notice

Hex-Ascii Encoder Decoder

Simple Tools used by students and programmers Worldwide.

Use this encode/decode online tool to encode and decode a string, binary, decimal, hexadecimal, words, password or phrases, and a few more useful for programmers and web dev
entities (special characters) name codes and number codes. See below for list and instructions. More Coming Soon! or you can contact us and make a suggestion. Back to busi

Ascii to Hexadecimal- encodes/ decode, (Ascii/text to hexadecimal string).

Encoder/Decoder

Select from 21

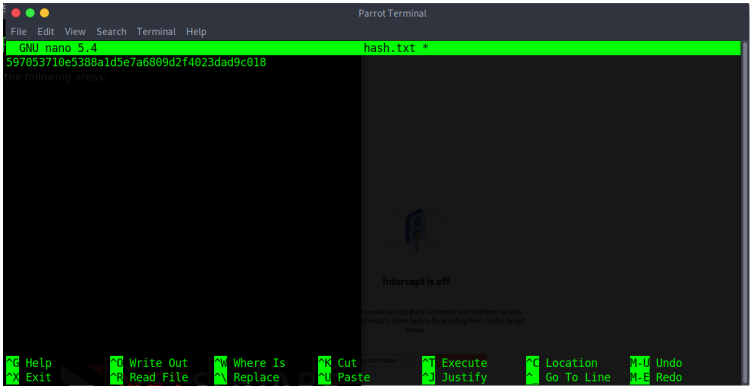
Hex-Ascii

Text Encode/Decode:

Net-Secure#2489040010#

EncodeDecode

- b. Use john and rockyou.txt to crack the hash



```
~$ john --wordlist=rockyou.txt hash.txt
Warning: detected hash type "Raw-SHA1", but the string is also recognized as "Raw-SHA1-AxCrypt"
Use the "--format=Raw-SHA1-AxCrypt" option to force loading these as that type instead
Warning: detected hash type "Raw-SHA1", but the string is also recognized as "Raw-SHA1-Linkedin"
Use the "--format=Raw-SHA1-Linkedin" option to force loading these as that type instead
Warning: detected hash type "Raw-SHA1", but the string is also recognized as "ripemd-160"
Use the "--format=ripemd-160" option to force loading these as that type instead
Warning: detected hash type "Raw-SHA1", but the string is also recognized as "has-160"
Use the "--format=has-160" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 128/128 SSE2 4x])
Warning: no OpenMP support for this hash type, consider --fork=5 to off
Press 'q' or Ctrl-C to abort, almost any other key for status
cassidy
?
lg 0:00:00.00 DONE (2022-10-22 15:25) 50.00g/s 77200p/s 77200c/s 77200C/s 'chopper..134679
Use the "--show --format=Raw-SHA1" options to display all of the cracked passwords reliably
Session completed
```

c. Base 64 + ECB mod and 128 bit key to decrypt

AES Online Decryption

Enter text to be Decrypted

umF0NEptUGGX8MZBNWnT8+GYxPKwgQUhNQ5CPDB4=

Input Text Format: Base64 CHex

Select Cipher Mode of Decryption

ECB

Key Size in Bits

128

Enter Secret Key used for Encryption

0ffff713370fff

Decrypt

AES Decrypted Output (**Base64**):

TrmVOLVNIY3VyZSM3NjEYMiJ3NDQxW==

Decode to Plain Text

Net-Secure#7612227441#

Challenge List

OK, let us begin with observation skill

A3:2017-Sensitive Data Exposure

Over the last few years, this has been the most common impactful attack. The most common flaw is simply not encrypting sensitive data. When crypto is employed, weak key generation and management, and weak algorithm, protocol and cipher usage is common, particularly for weak password hashing storage techniques. For data in transit, server-side weaknesses are mainly easy to detect, but hard for data at rest.

Use your powers of observation and identify algorithm for decode/dehashing/decrypt the following data.

In encryption section, we are using Symmetric-key algorithm with **ECB** mode and **128** sized key and key is "0ffff713370fff".

Encoded DATA : Vkd0a1ZrMXJOV1ZaZHEKT11sMkZ1R13ZY0ZwT1ZUvnhR13PVDFaSFRqV1V1MEJXW1ZVe5sU11SR1VsYAVZM1ZEQ1IMJazVWTVraU1d1Q1Bva1V4T1T5MMNfSmehkZMWkZoa1RtRn3NRGs9

Hashed DATA : 597053710e5388a1d5e7a6809d2f4023dad9c018

Encrypted DATA : umF0NEptUGGX8MZBNWnT8+GYxPKwgQUhNQ5CPDB4=

Encoded Value :


Hashing Value :

Encrypted Value :

submit

Challenges	Value	Result
Encoding	Net-Secure#2489040010#	passed
Hashing	cassidy	passed
Encryption	Net-Secure#7612227441#	passed

Yes you did! Continue on to the next challenge



secure • innovate • automate

2. Time traveller
Wayback machine to achieve old version

Can You Chip In?

Free and open access to public information is critical for any functioning democracy. The Internet Archive (which runs this project) is announcing the [Democracy's Library Initiative](#) to digitize and catalog important government documents—including laws, scientific reports, safety standards, patents, and so much more—so that citizens everywhere can be better informed.

You can help us accomplish this work. We build and maintain all our own systems, relying on the generosity of individuals to help us keep the record straight. We'd be deeply grateful if you'd join the 1 in 1000 users that support us financially. If you find our resources useful, [please pitch in](#).

Choose an amount (USD)

\$5

\$50

\$100

Custom: \$

☐ I'll generously add \$1.40 to cover fees.

☐ Make this monthly

Continue

Remind Me

Internet Archive

Wayback Machine

http://net-square.com/

331 captures

11 Apr 2001 - 9 Oct 2022

Go

APR 11 2001

MAY 2002

About this capture

Net SQUARE

Solutions for the E-Commerce Age

The Company

Service Lines

What's New

Contact Us

Mailing List

Site Home

Welcome to Net Square!

Information Technology, no longer an afterthought, is integral to a company's business strategy. Using it effectively would set a company on its way to realize its fullest potential. Ignoring it would be a costly inaction.

Net Square is designed to help empower your organization to do business on the new frontier, the Internet. If you organization is already doing business the electronic way, Net Square can help unlock its full potential and operate at maximum efficiency. We are committed to client service, and deliver solutions in accordance to the industry's leading practices.

Click on the menu items in the sidebar on the left to explore Net Square and see how you would like to be a part of the exciting field of electronic commerce! Enjoy your visit here at Net Square. Should you require further information on almost anything that you see, do not hesitate to drop us a line at info@net-square.com.

Mission Code

At the core, we are a technology based services organization.

Value Code

At Net Square, we strongly believe in sharing. We share our assets with clients, practice partners and employees by ethical business practices and entrepreneurial

Old is Gold! Find older data using Web Time Traveller

A3:2017-Sensitive Data Exposure

WSTG-INFO-01 Conduct Search Engine Discovery Reconnaissance for Information Leakage

WSTG-CONF-04 Review Old Backup and Unreferenced Files for Sensitive Information

"When the NetSquare site [net-square.com] launched in Apr. 2001, What was the banner/slogan on the website?"

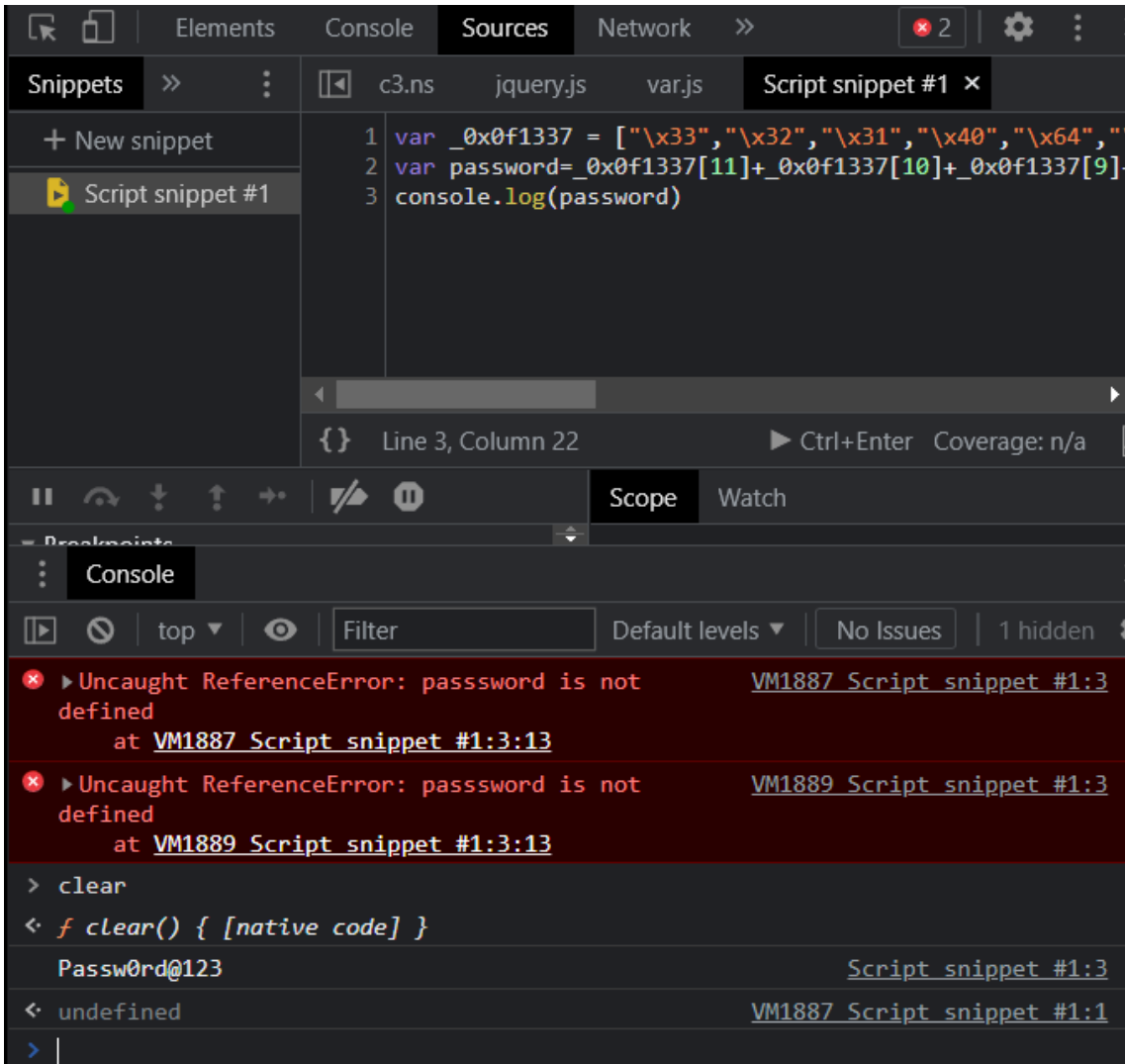
Slogan: Solutions for the E-Commel submit

Yes you did!

3. User Login Bypass

Insepcted code and find a js file var.js in /admin/ path

Run the code with a `console.log(password)`



Admin Login

Challenge List

A2:2017-Broken Authentication

WSTG-CONF-05 Enumerate Infrastructure and Application Admin Interfaces

The prevalence of broken authentication is widespread due to the design and implementation of most identity and access controls. Session management is the bedrock of authentication and access controls, and is present in all stateful applications. Attackers can detect broken authentication using manual means and exploit them using automated tools with password lists and dictionary attacks.

Login as admin user

User Name : admin

Password :

Submit

Yes you did! Continue on to the next challenge

4. Fingerprinting

Intercepted response in burpsuit to find details

Response from http://enigma.ctf.ns.exploitlab.net:80/web/c4.ns [162.252.242.82]

☒ Pretty ☐ Raw ☐ Hex ☐ Render

```
1 HTTP/1.1 200 OK
2 Date: Sat, 22 Oct 2022 10:13:43 GMT
3 Server: Microsoft-IIS/10.0
4 X-Frame-Options: SAMEORIGIN
5 X-Content-Type-Options: nosniff
6 Application-Engine: NS-ENG
7 Application-Server: Microsoft-IIS/10.0
8 Expires: Thu, 19 Nov 1981 08:52:00 GMT
9 Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
10 Pragma: no-cache
11 Vary: Accept-Encoding
12 Content-Length: 1312
13 Connection: close
14 Content-Type: text/html; charset=UTF-8
15
```

[Challenge List](#)

Fingerprinting

WSTG-INFO-02 Fingerprint Web Server

Web server fingerprinting is the task of identifying the type and version of web server that a target is running on. While web server fingerprinting is often encapsulated in automated testing tools, it is important for researchers to understand the fundamentals of how these tools attempt to identify software, and why this is useful.

Accurately discovering the type of web server that an application runs on can enable security testers to determine if the application is vulnerable to attack. In particular, servers running older versions of software without up-to-date security patches can be susceptible to known version-specific exploits.

What web server is this site running on?

Microsoft-IIS/10.0

What is the back-end application engine?

NS-ENG

5. HTTP Form Manipulation

Inspected the page and changed values in form

```

    <td>
      <input type="text" name="my_textfield" value="more than 15 chars" minlength=
        "15">
      <span style="color: red;">make this value more than 15 characters</span>
    </td>
  </tr>
  <tr>
    <td>Dropdown</td>
    <td>
      <select name="my_dropdown">
        <option value="redhat" selected>redhat</option></slot>
        <option value="whitehat">White Hat</option></slot>
        <option value="greyhat">Grey Hat</option></slot>
      </select>
      <span style="color: red;">make this "redhat"</span>
    </td>
  </tr>
  <tr>
    <td>Radio Button</td>
    <td>
      <input type="radio" name="my_radio" value="symbian" checked>
        "symbian"
      <input type="radio" name="my_radio" value="ios">
        "ios "
      <input type="radio" name="my_radio" value="windows8">
        "Windows 8 "
      <input type="radio" name="my_radio" value="blackberry">
        "BlackBerry "
      <span style="color: red;">make this "symbian"</span>
    </td>
  </tr>
  <tr>
    <td> == $0
    <input type="hidden" value="Hacked!" name="my_hidden">
  </td>
  </tr>
</tr>...</tr>
```

Challenge List

Play with HTTP Forms

A6:2017-Security Misconfiguration

Security misconfiguration can happen at any level of an application stack. Some time developer puts validation only on client side and missed validating parameter on server side.

Text Field

max 15 chars

make this value more than 15 characters

Dropdown

Black Hat

make this "redhat"

Radio Button

☒ Android ☐ iOS ☐ Windows 8 ☐ BlackBerry

make this "symbian"

Submit

Name	Value	Result
my_textfield	more than 15 chars	passed
my_dropdown	redhat	passed
my_radio	symbian	passed
Who am I?	Hacked!	passed

Level cleared! Next challenge

6. Client side input validation bypass

Intercepted the response in burpsuit and changed form validation script

Response from http://enigma.ctf.ns.exploitlab.net:80/web/c6.ns [162.252.242.82]

Forward Drop Intercept is on Action Open Browser

```
Pretty Raw Hex Render
9 function validateForm() {
10     var name = document.forms["moreformplay"]["my_name"].value;
11     var pass1 = document.forms["moreformplay"]["my_pass1"].value;
12     var pass2 = document.forms["moreformplay"]["my_pass2"].value;
13     var out_name = document.getElementById('out_name');
14     var out_pass1 = document.getElementById('out_pass1');
15     var out_pass2 = document.getElementById('out_pass2');
16     out_name.innerHTML = "";
17     out_pass1.innerHTML = "";
18     out_pass2.innerHTML = "";
19     if(!name.match("[a-zA-Z ]+")) {
20         out_name.innerHTML = "Alphabets only please!";
21         return true;
22     }
23     if(pass1 != pass2 || pass1 == "" || pass2 == "") {
24         out_pass1.innerHTML = "Passwords don't match";
25         return true;
26     }
27     else {
28         return false;
29     }
30 }
31 </script>
32 </head>
33 <body>
34 <h1>
35     More Fun With Forms
36 </h1>
37 <div style="color: green">
38     <b style="color: #90ee90">
39         A6:2017-Security Misconfiguration
40     </b>
41     <br>
42     Security misconfiguration can happen at any level of an application stack. Some time developer puts validation only on client side
43     and missed validating parameter on server side. <br>
44 </div>
45 <div id="score">
46     <a href="../score.ns">
47         Challenge List
48     </a>
49 </div>
50 <form name="moreformplay" method="get" onsubmit="return validateForm()">
51     <input type="text" value="Name" />
52     <input type="password" value="Pass1" />
53     <input type="password" value="Pass2" />
54     <input type="submit" value="Submit" />
55 </form>
```

Request to http://enigma.ctf.ns.exploitlab.net:80 [162.252.242.82]

Forward Drop Intercept is on Action Open Browser

```
Pretty Raw Hex
1 GET /web/c6.ns?my_name=1234&my_pass1=rcfvgh&my_pass2=rxdtcfvgh&my_submit=Submit HTTP/1.1
2 Host: enigma.ctf.ns.exploitlab.net
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Hacker/1.0
5 Accept:
6 text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
7 Referer: http://enigma.ctf.ns.exploitlab.net/web/c6.ns
8 Accept-Encoding: gzip, deflate
9 Accept-Language: en-US,en;q=0.9
10 Cookie: PHPSESSID=14p6bf04v8anu8jk215pd39gr7; __utmc=3bab5c84e0b9c3018301b37a0a31363c; __utmp=3bab5c84e0b9c3018301b37a0a31363c; __utme=3bab5c84e0b9c3018301b37a0a31363c; __utma=3bab5c84e0b9c3018301b37a0a31363c
11 Connection: close
12
```

Challenge List

More Fun With Forms

A6:2017-Security Misconfiguration
Security misconfiguration can happen at any level of an application stack. Some time developer puts validation only on client side and missed validating parameter on server side.

Enter Your Name

Only Alphabets

Deliberately submit non-alphabet characters

Choose A Password

Deliberately make the passwords mismatch

Verify The Password

Your Browser

Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36

Change to "Hacker/1.0"

Submit

Submit the form using GET instead of POST

Name	Value	Result
Character Restriction Bypass	1234	passed
Password Verification Bypass	rcfvgh rxdctfvgb	passed
User Agent Modification	hacker/1.0	passed
HTTP Form Submission method	GET	passed

Now that you are warmed up, dive straight into another test.

Click here to continue

**NETSQUARE**
secure • innovate • automate

7. User ID enumeration

Used burpsuit intruder and bruteforced UID field

Request	Payload	Status	Error	Timeout	Length	*NS...	Comment
1000	1999	200	<input type="checkbox"/>	<input type="checkbox"/>	1768	2	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	1762	1	
1	1000	200	<input type="checkbox"/>	<input type="checkbox"/>	1761	1	
2	1001	200	<input type="checkbox"/>	<input type="checkbox"/>	1759	1	
3	1002	200	<input type="checkbox"/>	<input type="checkbox"/>	1764	1	
4	1003	200	<input type="checkbox"/>	<input type="checkbox"/>	1762	1	
5	1004	200	<input type="checkbox"/>	<input type="checkbox"/>	1759	1	
6	1005	200	<input type="checkbox"/>	<input type="checkbox"/>	1763	1	
7	1006	200	<input type="checkbox"/>	<input type="checkbox"/>	1758	1	
8	1007	200	<input type="checkbox"/>	<input type="checkbox"/>	1758	1	
9	1008	200	<input type="checkbox"/>	<input type="checkbox"/>	1760	1	

Request

Response

Pretty

Raw

Hex

Render

User Enumeration

A5:2017-Broken Access Control
Access control weaknesses are common due to the lack of automated detection, and lack of effective functional testing by applying automated static or dynamic testing. Manual testing is the best way to detect missing or ineffective access control, including HTTP methods.

Enumerate user list and find "NS-ADMIN" user's Password.

Application have 2000 user, Your User ID is 1999 : User Name is "NS-ADMIN" and Password is "WhAti5MyPa55"

Password of NS-ADMIN : asdd

submit

Try again

Challenge List

User Enumeration


A5:2017-Broken Access Control
Access control weaknesses are common due to the lack of automated detection, and lack of effective functional testing by application developers. Access control detection is not typically amenable to automated static or dynamic testing. Manual testing is the best way to detect missing or ineffective access control, including HTTP method (GET vs PUT, etc), controller, direct object references, etc.

Enumerate user list and find "NS-ADMIN" user's Password.

Application have 2000 user, Your User ID is 1441 : User Name is "CAMERON" and Password is "2222"

Password of NS-ADMIN :

Yes you did! Continue on to the next challenge



8. 2FA bypass

Challenge List

OTP Bruteforce

A2:2017-Broken Authentication
The prevalence of broken authentication is widespread due to the design and implementation of most identity and access controls. Session management is the bedrock of authentication and access controls, and is present in all stateful applications. Attackers can detect broken authentication using manual means and exploit them using automated tools with password lists and dictionary attacks.

Try to understand client side operation and find 3 digit OTP using bruteforce

OTP :

Yes you did! Continue on to the next challenge

9.HTML5 Storage

Application

- Manifest
- Service Workers
- Storage

Storage

- Local Storage
 - <http://enigma.ctf.ns.exploitlab.com/>
- Session Storage
- IndexedDB
- Web SQL
- Cookies
 - <http://enigma.ctf.ns.exploitlab.com/>
- Trust Tokens
- Interest Groups

Cache

- Cache Storage
- Back/forward cache

Background Services

Filter

Key	Value
__utma	3bab5c84e0b9c3018301b37a0a31363c
NS_KEY	Y93xibcFwYWHXr7N2J5Atxo4iEWiU35n

1 Y93xibcFwYWHXr7N2J5Atxo4iEWiU35n

Console Issues

Application

- Manifest
- Service Workers
- Storage

Storage

- Local Storage
 - <http://enigma.ctf.ns.exploitlab.com/>
- Session Storage
 - <http://enigma.ctf.ns.exploitlab.com/>
- IndexedDB
- Web SQL
- Cookies
 - <http://enigma.ctf.ns.exploitlab.com/>
- Trust Tokens
- Interest Groups

Cache

- Cache Storage
- Back/forward cache

Filter

Key	Value
__utma	3bab5c84e0b9c3018301b37a0a31363c
NS_KEY	tvBpfjMLXFFUnMTToKxD4BAhdG0tSrFVN

1 tvBpfjMLXFFUnMTToKxD4BAhdG0tSrFVN

Application

Manifest

Service Workers

Storage

Storage

Local Storage

http://enigma.ctf.ns.exploitl

Session Storage

http://enigma.ctf.ns.exploitl

IndexedDB

NS_DB - http://enigma.ctf.ns

nsdb

Web SQL

NS_DB

NS_KEY

Cookies

http://enigma.ctf.ns.exploitl

Trust Tokens

Interest Groups

Visible columns

HTML5 Storage

A3:2017-Sensitive Data Exposure

Over the last few years, this has been the most common impactful attack. The most common flaw is simply not encrypting sensitive data. When crypto is employed, weak key generation and management, and weak algorithm, protocol and cipher usage is common, particularly for weak password hashing storage techniques. For data in transit, server-side weaknesses are mainly easy to detect, but hard for data at rest.

Retrieve the HTML5 Storage/Communication and get the keys, which is stored in client side.

Local Storage NS_KEY:	Y93xibcFwYWHXr7N2J5Atxo4iEWiU35n
Session Storage NS_KEY:	tvBpfjMLXFFUnMTToKxD4BAhdG0tSrFVN
IndexedDB NS_KEY:	7GhHBQpSDdyxp5XRASosupITGsuyiHT
Web SQL NS_KEY:	PzuR7IHlaHfg8UDkHtarbVZyXDUbp0sc

Name	Value	Result
Local Storage NS_KEY	Y93xibcFwYWHXr7N2J5Atxo4iEWiU35n	passed
Session Storage NS_KEY	tvBpfjMLXFFUnMTToKxD4BAhdG0tSrFVN	passed
IndexedDB NS_KEY	7GhHBQpSDdyxp5XRASosupITGsuyiHTL	passed
Web SQL NS_KEY	PzuR7IHlaHfg8UDkHtarbVZyXDUbp0sc	passed

Yes you did!



NETSQUARE

secure • innovate • automate

Decoded the jwt token



Hint : 256-bit-secret is "secret" with HS256 algo. Server required "login":"1"

Login as admin user

Password :

Submit

Yes you did! Continue on to the next challenge

13. Weak Session ID

Guessable Session ID

A2:2017-Broken Authentication
The prevalence of broken authentication is widespread due to the design and implementation of n is present in all stateful applications. Attackers can detect broken authentication using manual me

Application have 9999 user, You need to find user session of user id **8583** :

Session ID of user ID **0001** : abcdefgh-01234567-01234567-00011337
Session ID of user ID **0003** : cdefghij-23456789-23456789-00031337
Session ID of user ID **0005** : efghijkl-45678901-456789ab-00051337
Session ID of user ID **0014** : nopqrstu-34567890-def01234-00141337
Session ID of user ID **0015** : opqrstuv-45678901-ef012345-00151337
Session ID of user ID **0016** : pqrstuvw-56789012-f0123456-00161337
Session ID of user ID **0017** : rstuvwxy-67890123-01234567-00171337
Session ID of user ID **0018** : rstuvwxy-78901234-12345678-00181337
Session ID of user ID **0024** : xyzabcde-34567890-789abcde-00241337
Session ID of user ID **0025** : yzabcdef-45678901-89abcdef-00251337
Session ID of user ID **0026** : zabcdefg-56789012-9abcdef0-00261337
Session ID of user ID **0027** : abcdefgh-67890123-abcdef01-00271337
Session ID of user ID **0028** : bcdefghi-78901234-bcdef012-00281337
Session ID of user ID **9999** : opqrstuv-89012345-ef012345-99991337

Session ID of user ID 8583:

Yes you did! Continue on to the next challenge

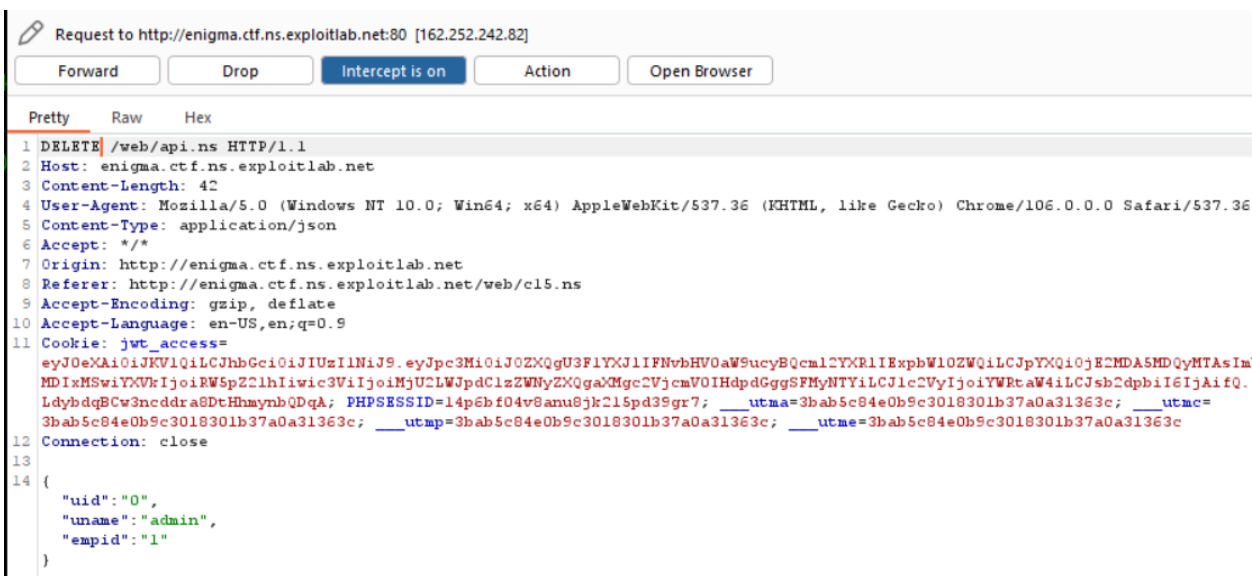
14. Insecure Direct Object Reference

1860	c14.ns:31			
1861	c14.ns:31			
1862	c14.ns:31			
1863	c14.ns:31			
1864	c14.ns:31			
1865	c14.ns:31			
1866	c14.ns:31			
1867	c14.ns:31			
1868	c14.ns:31			
1869	c14.ns:31			
1870	c14.ns:31			
1871	c14.ns:31			
1872	c14.ns:31			
1873	c14.ns:31			
1874	c14.ns:31			
1875	c14.ns:31			
1876	c14.ns:31			
1877	c14.ns:31			
1878	c14.ns:31			
1879	c14.ns:31			
1880	c14.ns:31			
1881	c14.ns:31			
<table border="1"><tr><td>User ID : 1866</td></tr><tr><td>User Name : JOSEF</td></tr><tr><td>User Password: <input type="password" value="Pussy1"/></td></tr></table>		User ID : 1866	User Name : JOSEF	User Password: <input type="password" value="Pussy1"/>
User ID : 1866				
User Name : JOSEF				
User Password: <input type="password" value="Pussy1"/>				
1882	c14.ns:31			
1883	c14.ns:31			
1884	c14.ns:31			
1885	c14.ns:31			
1886	c14.ns:31			
1887	c14.ns:31			
1888	c14.ns:31			
1889	c14.ns:31			
1890	c14.ns:31			
1891	c14.ns:31			
1892	c14.ns:31			
1893	c14.ns:31			
1894	c14.ns:31			
1895	c14.ns:31			
1896	c14.ns:31			
1897	c14.ns:31			
1898	c14.ns:31			
1899	c14.ns:31			
1900	c14.ns:31			

WSTG-ATHZ-04 Testing for Insecure Direct Object References

Yes you did! Continue on to the next challenge

15. HTTP method



REST API HTTP Methods

A5:2017-Broken Access Control

Access control weaknesses are common due to the lack of automated detection, and lack of effective functional testing by application developers. Access control detection is not typically amenable to automated static or dynamic testing. Manual testing is the best way to detect missing or ineffective access control, including HTTP method (GET vs PUT, etc), controller, direct object references, etc.

REST APIs enable you to develop any kind of web application having all possible CRUD (create, retrieve, update, delete) operations.

We have provide insert data forms to the user, but some how attacker is able to delete another user data. your task is to **delete admin user with emp id is 1 and uid is 0.**

Insert Your Data.

Enter Name :

Enter EID :

Yes you did!

Update Your Data.

Enter Name :

Enter EID :



Find Organisation Information

A3:2017-Sensitive Data Exposure

WSTG-INFO-01 Conduct Search Engine Discovery Reconnaissance for Information Leakage

WSTG-CONF-04 Review Old Backup and Unreferenced Files for Sensitive Information

Net Square has been launched on which date ?

Date (DDMMYYYY) :

Yes you did!



17. SSRF

Challenge List

Finance Management System

A6:2017-Security Misconfiguration
Security misconfiguration can happen at any level of an application stack, including the network services, platform, web server, application server, database, frameworks, custom code, and pre-installed virtual machines, containers, or storage. Automated scanners are useful for detecting misconfigurations, use of default accounts or configurations, unnecessary services, legacy options, etc.

We are using multiple internal servers to communicate internally application service.
our internal IP range is **192.168.0.1/22** and internal service port number is **8080** which contains **secret.txt** file.

Connect to Server

Server URL:

Token: **NS-6DC792B2900F5312A76B**

Submit Secret

Secret Token:

Challenge List

Finance Management System

A6:2017-Security Misconfiguration
Security misconfiguration can happen at any level of an application stack, including the network services, platform, web server, application server, database, frameworks, custom code, and pre-installed virtual machines, containers, or storage. Automated scanners are useful for detecting misconfigurations, use of default accounts or configurations, unnecessary services, legacy options, etc.

We are using multiple internal servers to communicate internally application service.
our internal IP range is **192.168.0.1/22** and internal service port number is **8080** which contains **secret.txt** file.

Connect to Server

Server URL:

Submit Secret

Secret Token:

Yes you did!

40. IP white listing bypass

Code Review : IP whitelisting Bypass

A2:2017-Broken Authentication

The prevalence of broken authentication is widespread due to the design and implementation of most identity and access controls.

Understand the code and bypass the local IP verification to complete the task.

Code snippet

```
function getIpAddress()
{
    return isset($_SERVER["HTTP_X_FORWARDED_FOR"]) ? $_SERVER["HTTP_X_FORWARDED_FOR"] : $_SERVER["REMOTE_ADDR"] ;
}

function isIpLocal()
{
    if(substr(getIpAddress(),0,4) == "127.")
    {
        return true;
    }
}

if(isset($_POST['value']))
{
    if(isIpLocal())
    {
        echo "Yes you did!";
    }
    else
    {
        echo "Try again. Your IP address is " . getIpAddress() ;
    }
}
```

Click on following button to send request to the server.

Yes you did! Continue on to the next challenge



NETSQUARE
secure • innovate • automate

Code:

gdnayoMs/abbBdZw0qdecslJDIvMaj9O5/5S56DNP/scZnY9O0/yDeroehh5sHbpUOYSYj7EE
NECA5vtPV4iTGsDXXO8sGc0nM9EZQiduW6pvCtXjMCOyxqAHBpxrl8MxWgJoIWpPCloyKs1vl
IhbQdPHulxntvi2qura7fj+shWmICGwtgg+UJRXmb3Mq1cnmg9qSbG9mEfpUEqK7drkiQ40ES
5uLnQUYNOMzOK9sV0ya3hpABq/sRLiz8pJlcgL6ayZe6mR/U3ZAnldNrx6C/B5tesKK2h/lbG
D7bQTbGJoRp/6nScZhMU6RImmhiFzYk7NRLO2nzqrWSIIXlpI7XPzTXV5vTmt9f8y49/LFaZz
KGvUVBwUwcQe4/g795sEvDI5ePcCoEkWk47Lk+wbF/ZWPNw8Iu7mFx6jVA8vQhHrSchhqXAgU
WSDzk322bz/Ce6aQ4tBpGgeH9Ak3BwmQwmYAcEEemQn1MTClzgpkHnaB5eaWIEK2eiDeeasq9z
LUG4MyBMYQhZI6cGd7wa0lgZgGKqNWyD4WjovLD/wkYmAun/mTFESMXYKXQrue3yhs8VZKBai
XXwaDG6NzF51J6yidcR+FyJ+4Dia+iwCWZVjNmnMtCbJVHjG/DK9EsG4lDDuJRmli16PUo34m
9rKvruj4aCwe3UupfEuXCGvDQL/uwZsXWfrq+ef52qWNFj23ATDNTlcFYOhxgauqRePg/ZQHQ
H5ilpc6Bo7/m4LlazzkV8MvPxkfrcnU2HFqQM3LUyBHPitpL2kjHhPrbTflVzHCFwgmqBat+34
EKoN+m3KLjYDDUFOHU5oMICoiHLuosN7/fgajzOdtqYB1W2qJgR9nG+u+jRcmVJHE/QxRWI5C
8Ac43SN24Ynrle2uA4c4PX0xrX5fmCdviLqBwRFPpg15SmuElvq6h9iHZV5AX78scQWO4gnu2
VyAX7J4nov/wqzrMUrK93Cp3KdksvjIW1kCwllWntOk0ed8Vi9Bypyl/Utji+GLxPGwsClwkt
0v6UbeiesOiCn5F0bDIZCDICceFWSYYAyQF7OubHmszCoOlCY9gN1td4u7IpwBkIK8xYW3UXT
zNHQyjMeycmguExfnesXRgwaX47+eLlEeDQZRA8zYOiZpsSiTOtP/7Qsgmat+Xs5xhQhJmM2R
b6CW/5fM58td2G+J8qgalokg6F4DVpFsaU92qGpR+FF5Pj7wD4bRYOTVkpSkNiNfzdZnPxPIg
c25xZ/HKX/T3ffh1Zbi7zW2YEJXVXcJUoAhdXZP/m685PjgnPHMgxf+V6YUFegWNUqQ9ezNUW
hjhTRvtvFbGD8TznNdGEdoapExbJztZDYqoDoTD8W9luUX9whixc9VLisuydf0cK1+fVfxV6R
YltziLJmlilKo4yAv4CmNr8XWFvutQfOfADQ0GCMhmeXwb+WD3eI36mFBLjP7086MyDwN208r
hyFq+Ps88LQUu6eZLKyrjxyj3t6n42oGysLiEP2UE3X6fRLoZ1Bm4HDUMmaRxv8rumGJpXNLE
GzFMEUcR7Y0hCuMdmVuVsOPYg==