<https://player.vimeo.com/video/286282599> Around 17 min

Mail Services 
• Simple Mail Transfer Protocol 
• Defined in RFC 821 
• Numerous amendments and additional RFC's over the years 
• SMTP services for a domain exist in MX records 
• MX records determine the priority for e-mail services 
• Lowest number gets highest priority 
• SMTP is one of the most widely used protocols on the Internet 
• Operates most commonly over TCP port 25 

Mail Service Ports 
• Mail services are also found on other ports 
• TCP 2525, 465 or 587 are all widely used 
• Email transport layer can be encrypted with SSL 
• SSL is provided on the socket or in the STARTTLS command 
• IMAP and POP3 provide remote mailbox services 
• IMAP runs on port 143 and/or port 993 
• POP3 runs on port 110 and/or port 995 — POP2 on port 109 
• Mail welcoming banners typically contain software details 

E-Mail Chain 
• Mail User Agent (Your email client, outlook / thunderbird) 
• Mail Transfer Agent (Your SMTP server, sendmail / exim) 
• Mail Delivery Agent (A more feature-rich mail software) 
MUA 
S MTp 
MTA 
MOA 
POP i' IMAP 
MUA 

Mail OpenRelay & 
User Enumeration 
• SMTP protocol can be used to enumerate usernames 
and/or valid email addresses 
• EXPN command 
• VRFY command 
• RCPT TO: command 
• Supplying these can often help expose common UNIX 
user accounts on UNIX MTA's 
• Delivering mail to arbitrary domains can allow the SMTP 
service to be abused for sending SPAM. 

DSN Reports 
• Delivery Status Notification Reports 
• These can be gathered from an organization by simply sending 
an email to an Invalid address. 
• The SMTP returned e-mail message can contain sensitive 
information 
• Headers will contain the IP addresses of the SMTP services 
used (MTA, MDA, MUA.) 
• Headers can also contain information on anti-virus software in 
use. 
• Can also be used to identify valid email addresses. 

Remote Mailboxes 
• POP services often do not honor account lockout policy 
• This service is usually susceptible to brute-force 
• Account lockouts are more common on Windows provided 
services (Exchange) and IMAP 
• WebMail is widely used and can also be susceptible to common 
web application layer security weaknesses 
• SSL vulnerabilities can also apply to mail services when 
enabled 
• Heartbleed on mail services can expose credentials 

Sendmail Vulnerabilities 
• Sendmail has a history of useful but old vulnerabilities 
• CVE-2006-0058 — Remote signal handling bug 
• CVE-2003-0161 — Remote prescan() code execution 
• Some even pre-date CVE's! 

520 
321 
322 
523 
324 
525 
326 
527 
328 
529 
530 
331 
532 
i 34 
Sendmail wizard?! 
• Sendmail had backdoors.. 
• Non-existent today 
CMDDBGWIZ: 
If (WizWord NULL) 
char seed[3] ; 
extern char •crypt(); 
(void) strncpy(seed, WizWord, 2); 
If (strcmp(WizWord, crypt(p, seed)) 
-Conmands : 
214 
214- 
HELO 
214- 
'OOP 
214-. 
214 
214- . 
214 
MAIL 
QUIT 
RCPT 
HELP 
DATA 
VRFY 
RSET 
EXPN 
-For more info use 
End of HELP info 
showq 
Send 
kill 
"HELP . 
Iswiz - TRUE; 
message( "200" , 
"Please pass, oh 
500 Mere mortals musn't mutter that mantra 
WIZ test 
500 You are no wizard! 
WIZ wizard 
e) 200 Please pass, oh mighty wizard 
kill 
200 Mother is dead 
mighty wizard"); 
"You are no wizard! " ) ; 

Exim 
• Exim is also widely used and has numerous vulnerabilities 
Remote string_format heap overflow 
• CVE-20104345 - 
• CVE-2010-4344 - 
privilege escalation 
• CVE-2015-0235 - GHOST libc() exploit 
• CVE-2016-1531 - 
privilege escalation 

CVE-2014-0160 Heartbleed 
TIS Heartbeat Request 
Heartbeat 
Payload Length 
Attac:W 
Paykwl 
OxFFFF byte 
Mhinun 16 
Payload length greater than his PaybXS. 
Sevke ser-as bwk abocated to try) Attack«s htvth. 
A w•éd appous! 

[](https://www.ethicalhacker.net/)

Focus areas might include:

* Web application penetration testing
* Network penetration testing
* Open-source Intelligence (OSINT)
* Digital Forensics & Incident Response (DFIR)
* Reverse engineering (RE)
* Malware research
* Physical penetration testing

One of the best tools I’ve found from the OWASP project is the [Juice Shop project](https://www.owasp.org/index.php/OWASP_Juice_Shop_Project). Created and managed by [Björn Kimminich](https://twitter.com/bkimminich), Juice Shop is a purposely vulnerable web application that you can run in a virtual machine and probe to find common vulnerabilities. It’s built on Node.js, the Express framework and AngularJS offering a modern web app architecture to practice both attacks and mitigation techniques. It’s one of my favorite projects, because it forces me to dive deeper into things like network requests, HTTP headers and API calls to break the app and score points using tools like the included browser developer tools as well as proxies like Burp Suite. There are plenty of other prepackaged vulnerable web apps ([Damn Vulnerable Web Application (DVWA)](http://www.dvwa.co.uk/)) and repositories ([Hack.me](https://hack.me/)), but, being my first, Juice Shop deserves special recognition.

<https://wipro.udemy.com/the-web-developer-bootcamp/learn/v4/t/lecture/3861190?start=60>

|  |  |
| --- | --- |
| <!DOCTYPE html> | To begin with |
| <html>x</html> | Start the code |
| <title>x</title> | Page Title |
| <head>x</head> | Heading for Page Title |
| <body>x </body> | Body for the words inside webpage |
| <h1>x</h1>  <h2>x</h2>  <h3>x</h3>  <h4>x</h4>  <h5>x</h5>  <h5>x</h5> | Heading    h3 and h4 are most commonly used |
| <b>x </b> | Make Bold |
| <p>x </p> | Paragraph |
| <a>x</a> | To create other links |
| <link>x</link> | To link a document with its related resources |
| <script>x</script> | To link a document with its related resources |
| <img>  <img src="\*location" alt="My test image"> | To display Image |
| <video>x</video> | To display Video |
| <audio>x</audio> | To display Audio |
| <iframe>x</iframe> | To display other HTML Documents |
| <strong>x</strong> | To make a word strong |
| <ul>x</ul>  <ol>x</ol>  <li>x</li> | Unordered list  Ordered list  List items |
| <a href="\*location">Mozilla Manifesto</a> |  |

<https://developer.mozilla.org/en-US/docs/Learn> - A Google for Web Development

<https://developer.mozilla.org/en-US/docs/Glossary> - A place for all abbreviations

<https://howdns.works/> - How DNS works by art and easy explanation

<https://dnsimple.com/trusty> - How Name Server was born and managed by art and easy explanation

<https://howhttps.works/> - How HTTPS works by art and easy explanation