Applied Cryptography in Cyber Security

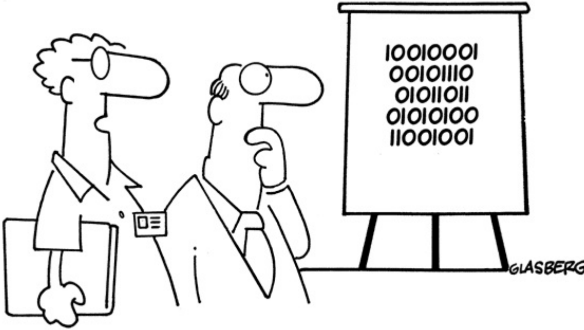
Jothi Ramalingam

Information Security

Safe-guarding the information from unauthorized access or modification

Principles

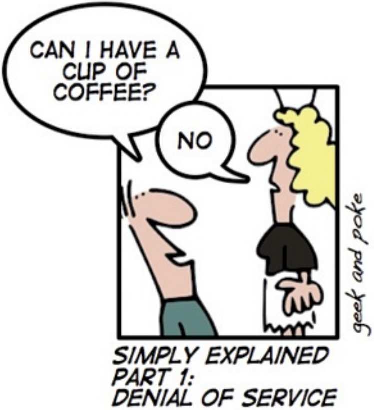
- Confidentiality : protecting data from unauthorised users
- Integrity: ensures that the data is not altered or deleted
- Availability: Information systems must be available to authorized entities when they need to access or use them



“We’ve devised a new security encryption code. Each digit is printed upside down.”

Availability

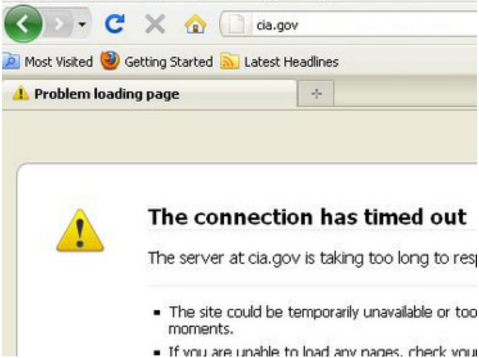
- Who needs Confidentiality and Integrity if the authorised users of information cannot access and use it?
- Loss of availability is often referred to as "Denial-of-Service"


CAN I HAVE A CUP OF COFFEE?
NO
geek and poke
SIMPLY EXPLAINED PART 1: DENIAL OF SERVICE

Recent DoS attacks

Central Intelligence Agency (Feb. 2012)

US, UK govt. Central Intelligence Agency (CIA) websites went offline due to a DoS attack



The connection has timed out

The server at cia.gov is taking too long to res

- The site could be temporarily unavailable or too moments.
- If you are unable to load any pages, check you

(Source: The Telegraph)

Recent Cyber attacks

South Korea (March 2011)

Some 29 institutions were affected by a type of DoS attacks.

Government ministries, the National Assembly, the military headquarters, US Forces in Korea and major banks were among those hit.

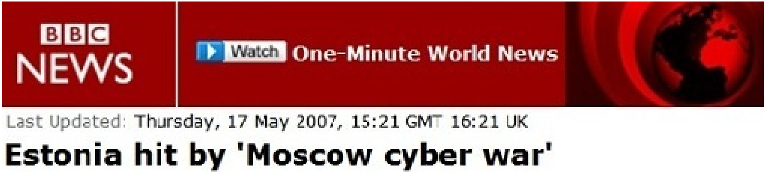
South Korea hit by cyber attacks

04 March 11 10:40 GMT

Recent Cyber attacks

Nation states:

Estonia (April 2007); Georgia (August 2008); United States and South Korea (July 2009).



Last Updated: Thursday, 17 May 2007, 15:21 GMT 16:21 UK

Estonia hit by 'Moscow cyber war'

Google (June 2009)

On June 25, 2009, the day Michael Jackson died, the spike in searches related to Michael Jackson was so big. For about 25 minutes, when some people searched Google News they saw a “We’re sorry” page.

Recent Cyber attacks

DDoS Attack

- GitHub, a software development platform (2018)
  - 1.35 Tbps
  - largest of its kind in History
- Dyn, a DNS service provider to major websites (Oct 2016)
  - mounted by a botnet Mirai connecting 100000 IoT's
  - massive load against DNS server (1.2 gbps)

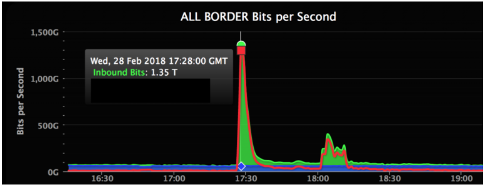


Image:https://www.welivesecurity.com/2016/10/24/10-things-know-october-21-iot-ddos-attacks/

Denial-of-service (DoS) attacks

Aim : To disrupt the availability of information systems and prevents legitimate users from accessing it.

- Brute force attacks: attacker generates sufficiently many legitimate-looking requests to overload a server's resources. Does not require special knowledge of protocol specification or implementation.
  - Distributed denial of service (DDoS) attacks
- Semantic attacks: attacker tries to exploit vulnerabilities of particular network protocols or applications. Requires special knowledge of protocol specification and implementation.
  - TCP SYN flooding / IP spoofing attacks

Now-a-days, DoS attacks against sites of your choice are readily available for hire.

TCP SYN flooding: An example semantic DoS attack

The Transmission Control Protocol (TCP)

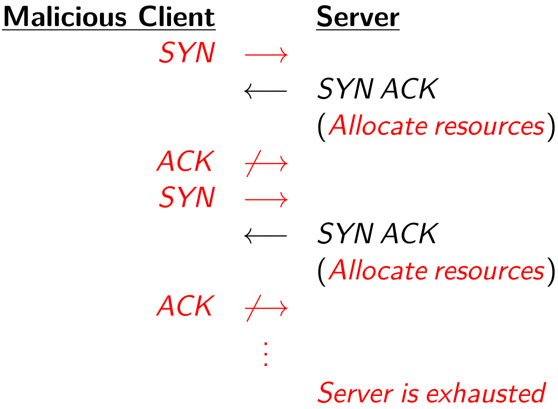
- One of the two main components of the Internet Protocol Suite (commonly referred to as TCP/IP)
- The TCP three-way handshake is the procedure used to establish or open a connection

TCP Normal

Client		*Server*
SYN	$\longrightarrow$	
	$\longleftarrow$	*SYN ACK*
		(*Allocate resources*)
ACK	$\longrightarrow$	

TCP SYN flooding (Contd.)

An attacker floods the server with SYN messages and leaves the protocol. The server's memory resource will soon be exhausted and no new connections (legitimate or not) can be made, resulting in DoS.



Malicious Client **Server**

SYN →

← *SYN ACK*

(*Allocate resources*)

ACK ↛

SYN →

← *SYN ACK*

(*Allocate resources*)

ACK ↛

⋮

Server is exhausted

DoS Attacks in Key Establishment Protocols

Goals of Key Establishment

Use cryptographic techniques to

- Authenticate each other
- Share a secret key

Limitations

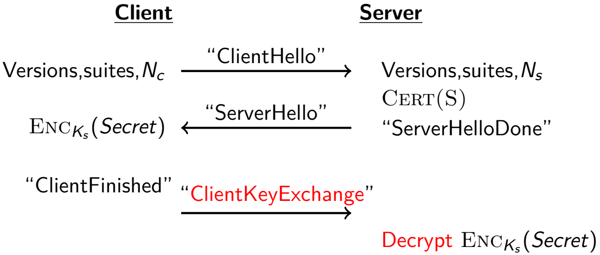
Involve computationally expensive operations such as modular exponentiation

- Vulnerable to a denial-of-service attack

“Core” SSL without client authentication

The most widely used and trusted protocol on the Internet.

The Secure Sockets Layer (SSL) protocol



Client

Server

Versions,suites,N_c $\xrightarrow{\text{“ClientHello”}}$ Versions,suites,N_s

$\textsc{Cert}(\mathrm{S})$

$\textsc{Enc}_{K_s}(Secret)$ $\xleftarrow{\text{“ServerHello”}}$ “ServerHelloDone”

“ClientFinished” $\xrightarrow{\text{“ClientKeyExchange”}}$

Decrypt $\textsc{Enc}_{K_s}(Secret)$

DoS vulnerabilities

- No DoS resilient features.
- Involves expensive public-key operations.
- This can be easily exploited by a DoS attacker.

How to mitigate DoS attacks?

Prevention techniques

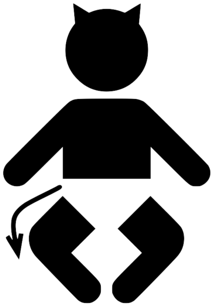
Try to identify malicious traffic:

- address filtering to block false addresses or addresses making too many requests;
- bandwidth management by routers and switches;
- packet inspection: look for patterns of bad requests;
- intrusion-prevention systems: look for signatures of attacks.

The Main Problem

Difficult to distinguish real users' legitimate requests from attacker's legitimately-formed requests in DoS attacks.

Legitimate User



Malicious User

(Designed by Sam Small)

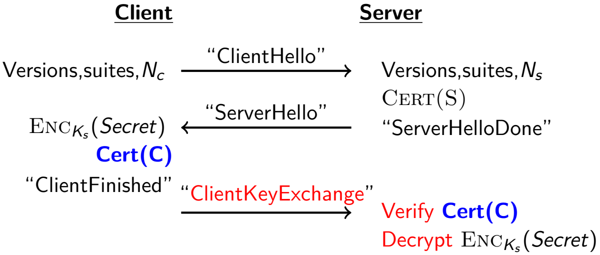
What can we do now?

💡 **Authentication**

“Core” SSL with client authentication

Authentication is a promising way but is a computationally intensive.

The Secure Sockets Layer (SSL) protocol



Client

Server

Versions,suites,N_c $\xrightarrow{\text{“ClientHello”}}$ Versions,suites,N_s

$\textsc{Cert}(\mathrm{S})$

$\xleftarrow{\text{“ServerHello”}}$ “ServerHelloDone”

$\textsc{Enc}_{K_s}(\mathit{Secret})$

Cert(C)

“ClientFinished” $\xrightarrow{\text{“ClientKeyExchange”}}$

Verify **Cert(C)**

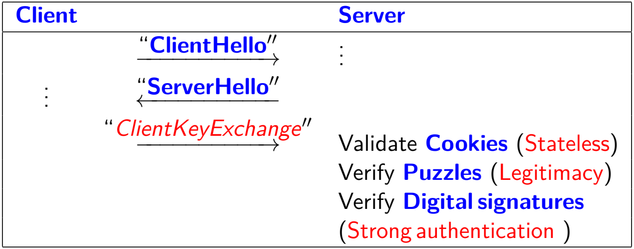
Decrypt $\textsc{Enc}_{K_s}(\mathit{Secret})$

DoS vulnerabilities

- Still involves relatively expensive signature operations.
- This can be easily exploited by a DoS attacker.
- DoS attacks cannot be prevented *completely* but can be **mitigated**.

Gradual authentication (Meadows, 2000)

- Idea is to use cheap and low-security authentication initially
- Gradually put more effort into authentication if earlier stages succeed
- A typical progression might be to implement cookies first, then puzzles, then strong cryptographic authentication.
  - **Cookies** provide proof of reachability
  - **Puzzles** provide proof of work
  - **Signatures** provide strong cryptographic authentication



Client **Server**

$\xrightarrow{\text{“ClientHello”}}$

$\xleftarrow{\text{“ServerHello”}}$

$\xrightarrow{\text{“ClientKeyExchange”}}$

Validate **Cookies** (Stateless)
Verify **Puzzles** (Legitimacy)
Verify **Digital signatures** (Strong authentication)

DoS-resistant strategies

Strategies		**Techniques**
1.Counterbalancing memory expenditure	→	Cookies
2.Counterbalancing computational expenditure	→	Client puzzles
3.Gradual authentication	→	Reordering protocol operations

Client puzzles or proof of work

- First presented by Dwork and Naor to combat Junk emails. Later the concept was extended by Juels and Brainard to combat DoS attacks
- Issued when a server is under attack
- Receiving puzzle solution guarantees the legitimate intentions of the client
- Small computational overhead for legitimate clients
- To flood the server by initiating enormous amount of connection requests, an attacker has to do more computationally expensive operations

Properties of good client puzzle

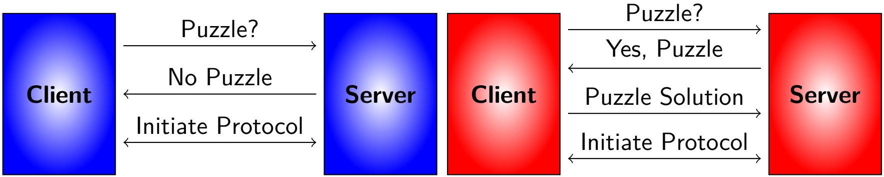
- Should be cheap to generate and verify for the server, but moderately hard to find the solution for the attacker
- The same client puzzle may be given to several clients.
- The difficulty in finding client puzzle solution can be adjusted to any level from zero to infinity (not solvable).

Client puzzles

Moderately-hard cryptographic problems sent back to clients who make requests.

- Client has to solve puzzle before receiving service from server.
- Puzzles should be easy to generate and verify.
- Puzzle difficulty can be adjusted from easy to hard.
- The cost is much higher for an attacker than for a legitimate client.

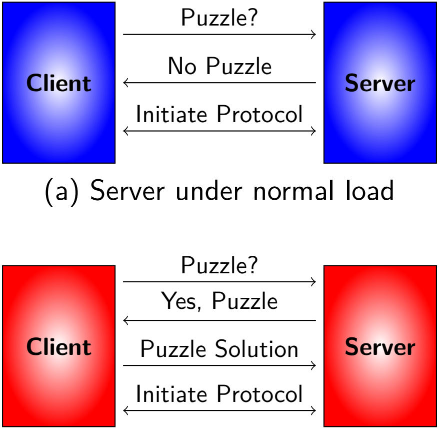
How do client puzzles work?



(a) Server under normal load

(b) Server under attack

How do client puzzles work?



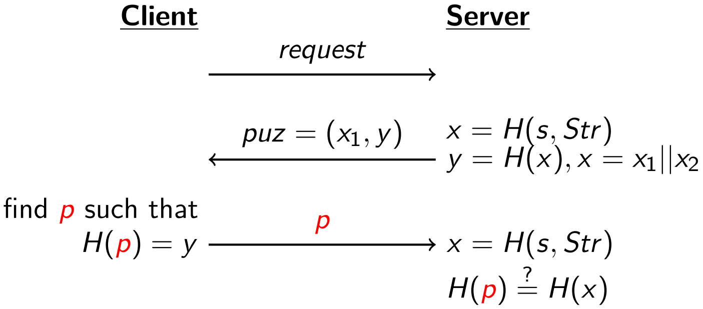
(a) Server under normal load

(b) Server under attack

Client puzzle: An example

Juels and Brainard (1999)

- Introduced the notion of client puzzles
- Based on the problem of partially inverting Hash functions



Client **Server**

request

$puz = (x_1, y)$

$x = H(s, Str)$

$y = H(x), x = x_1 || x_2$

find p such that $H(p) = y$

p

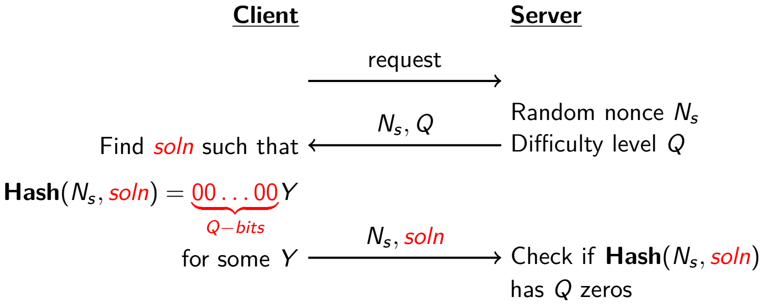
$x = H(s, Str)$

$H(p) \overset{?}{=} H(x)$

An example client puzzle

Aura *et al.* (2000)

- A puzzle scheme based on hash functions
- Puzzle: Finding a partial hash inversion by brute force search



Client

Server

request

Random nonce N_s

Difficulty level Q

N_s, Q

Find *soln* such that

$\textbf{Hash}(N_s, soln) = \underbrace{00\ldots00}_{Q-bits}Y$

for some Y

$N_s, soln$

Check if $\textbf{Hash}(N_s, soln)$ has Q zeros

Server: 1 Hash, Client: 2^Q Hashes, on avg.

The most efficient puzzle scheme in the literature.

Puzzle security properties

- Difficulty: it should be moderately hard to solve a puzzle (computation-bound or memory-bound)
- Unforgeability: it should not be possible for the adversary to generate valid puzzles
- Non-parallelizability: it should not be possible to have multiple computers solve a puzzle in less time than a single computer could
- Tuneable difficulty: can provide puzzles with different difficulty levels
- Useful puzzles: the work done in solving a puzzle can be used for another purpose

Defining DoS-resistant protocols

- Adversary controls communication between all parties.
- Adversary can gain server secret information via Expose query.
- Adversary can get clients to solve puzzles.
- The probability that an efficient adversary can make the server accept n puzzle instances should be bounded by a non-decreasing function $\epsilon_{k,n}(t)$ where $\epsilon_{k,n}(t) \leq \epsilon_{k,1}(t/n)$.
- Server should not perform expensive operations in a protocol run until puzzle is solved.

Mitigating DoS attacks with fast-to-verify measures

Two ways to improve DoS resistance in protocols

- **Increasing** client side cost
  - Client puzzles increase the cost of causing an attack.
  - Little bit overhead for the server.
- **Decreasing** server side cost
  - Replacing costlier protocol operations with lighter ones without compromising security of the protocol.
  - Improves the performance and DoS resistance.

“The world’s fastest digital signature scheme” (Bernstein, 2000)

Fast-Verification Digital Signature (FVDS) scheme

Signer

To sign a message M,
Generate,
$\textsc{Sign}(M) = (r, s, f, h, t, n)$
s.t. $h = \mathbf{Hash}(M, r)$
and $s^2 = f \cdot h + t \cdot n$

$$\xrightarrow{M, (r, s, f, h, t, n)}$$

Verifier

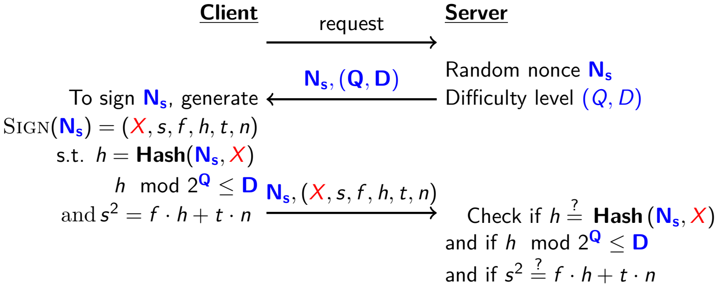
Check if $h \overset{?}{=} \mathbf{Hash}(M, r)$
and if $s^2 \overset{?}{=} f \cdot h + t \cdot n$

Here $n = p \cdot q$ is the public key and the pair (p, q) is the secret key.

- **Only** a few integer operations needed to verify a signature.
- Reduces the server side cost significantly if used in protocols.
- Can be more efficient if operations are done modulo a small prime.

Combined DoS countermeasure

FVDS-based client authentication with puzzles



Client

Server

request

Random nonce $\mathbf{N_s}$

Difficulty level (Q, D)

$\mathbf{N_s}, (\mathbf{Q}, \mathbf{D})$

To sign $\mathbf{N_s}$, generate

$\textsc{Sign}(\mathbf{N_s}) = (X, s, f, h, t, n)$

s.t. $h = \mathbf{Hash}(\mathbf{N_s}, X)$

$h \mod 2^{\mathbf{Q}} \leq \mathbf{D}$

and $s^2 = f \cdot h + t \cdot n$

$\mathbf{N_s}, (X, s, f, h, t, n)$

Check if $h \overset{?}{=} \mathbf{Hash}(\mathbf{N_s}, X)$

and if $h \mod 2^{\mathbf{Q}} \leq \mathbf{D}$

and if $s^2 \overset{?}{=} f \cdot h + t \cdot n$

Advantages

- **Only** one hash operation needed to verify a puzzle solution.
- This adds no extra cost as the server must otherwise compute it for signature verification.

Efficiency of FVDS

RSA vs. FVDS

	32-bit i386 build		
modulus (bits)	RSA $e = 65537$	FVDS	
		full verify	fast verify
1024	14013	112690 (8×)	79502
2048	3949	52036 (13×)	55838
4096	1013	20688 (20×)	42650

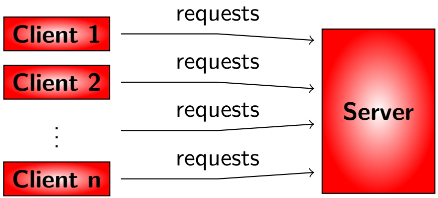
Table: 1. Signature verification performance in operations per second (OpenSSL 1.0.0 (modified), Intel Core 2 Duo 2.53GHz T9400, one core).

RSA w/ GPuz vs. FVDS w/ GPuz

	32-bit i386 build	
modulus (bits)	RSA ($e = 65537$) with GPuz	FVDS (full verify) with built-in puzzle
1024	13970	112690 (8×)
2048	3943	52036 (13×)
4096	1011	20688 (20×)

Table: 2. Performance of client authentication with puzzle in operations per second (OpenSSL 1.0.0 (modified), Intel Core 2 Duo 2.53GHz T9400, one core).

Attack simulation



Server under attack

Clients: multiple machines across a dedicated network with no other traffic or programs running.

- Modified OpenSSL to include support for a hash-based client puzzle and for the FVDS-based authentication protocol with built-in puzzle.
- Modified the Apache web server as needed to support these changes.
- Used the http_load package which can generate many client requests over either http or https (when used with OpenSSL);

Tests for SSL with new countermeasures

- **Test 1->“no puzzle”**: no puzzle is used. Used only RSA cipher suite and FVDS cipher suite.
- **Test 2 -> “hash:12, legitimate solutions”**: This test included our hash-based client puzzle with difficulty set to 12; the client needs to find a pre-image x such that the hash value H(x) starts with at least 12 zero bits (where H is the SHA-1 hash function).
- **Test 3->“fvds:12, legitimate solutions"**: This test, only for the FVDS-based cipher suite, is similar to Test 2 except that the hash-based puzzle is integrated with the FVDS signature generation/verification with $Q = 12$ and $D = 0$.
- **Test 4-> “hash:12 / fvds:12, mix legitimate/garbage"**: In this test, 100 legitimate clients are simulated, as well as a large number of attacking clients sending fake requests.

Performance results and observations

Key transport ⟶		**RSA-1024**	**RSA-1024**	**RSA-1024**
Client authentication ⟶		**none**	**RSA-1024**	**FVDS-1024**
Server configuration	**Client's puzzle strategy**	⇓	⇓	⇓
1: no puzzle		1924	(16% ↓) 1621	(10% ↓) 1732
2: hash:12	legitimate solutions	1911	1597	1719
3: fvds:12	legitimate solutions	N/A	N/A	1732
4: hash:12 / fvds:12	mix legitimate/garbage	100 legitimate 4302 garbage	100 legitimate 2767 garbage	100 legitimate (8% ↑) 3022 garbage

Table: Number of SSL connections per second.

Cyber attacks since COVID outbreak

Ransomware Attack

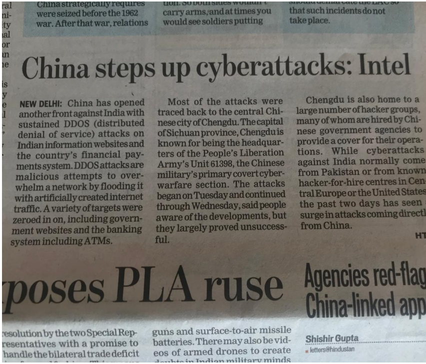
- files of victim are taken hostage and ransom to be paid for getting them back
- a software giant (not **TCS**) has become a victim
- Employer and employees are in fear of losing business and job respectively

Cyber attacks since COVID outbreak (Contd.)

Is China planning a war in Cyber Space?

DoS Attack on India

- Chinese hackers attempted 40,000 cyber attacks on Indian web, banking sector in 5 days (June 24, 2020)
- **Intel**: China opens another front, steps up cyberattacks that target India
- Country's power infrastructure could be the next target of terrorists looking to cripple its economy.



China strategically requires were seized before the 1962 war. After that war, relations

tion. So both sides wouldn't carry arms, and at times you would see soldiers putting

should demarcate the LAC so that such incidents do not take place.

China steps up cyberattacks: Intel

NEW DELHI: China has opened another front against India with sustained DDOS (distributed denial of service) attacks on Indian information websites and the country's financial payments system. DDOS attacks are malicious attempts to overwhelm a network by flooding it with artificially created internet traffic. A variety of targets were zeroed in on, including government websites and the banking system including ATMs.

Most of the attacks were traced back to the central Chinese city of Chengdu. The capital of Sichuan province, Chengdu is known for being the headquarters of the People's Liberation Army's Unit 61398, the Chinese military's primary covert cyberwarfare section. The attacks began on Tuesday and continued through Wednesday, said people aware of the developments, but they largely proved unsuccessful.

Chengdu is also home to a large number of hacker groups, many of whom are hired by Chinese government agencies to provide a cover for their operations. While cyberattacks against India normally come from Pakistan or from known hacker-for-hire centres in Central Europe or the United States the past two days has seen surge in attacks coming directl from China.

HT

poses PLA ruse

resolution by the two Special Representatives with a promise to handle the bilateral trade deficit

guns and surface-to-air missile batteries. There may also be videos of armed drones to create doubts in Indian military minds

Agencies red-flag China-linked app

Shishir Gupta

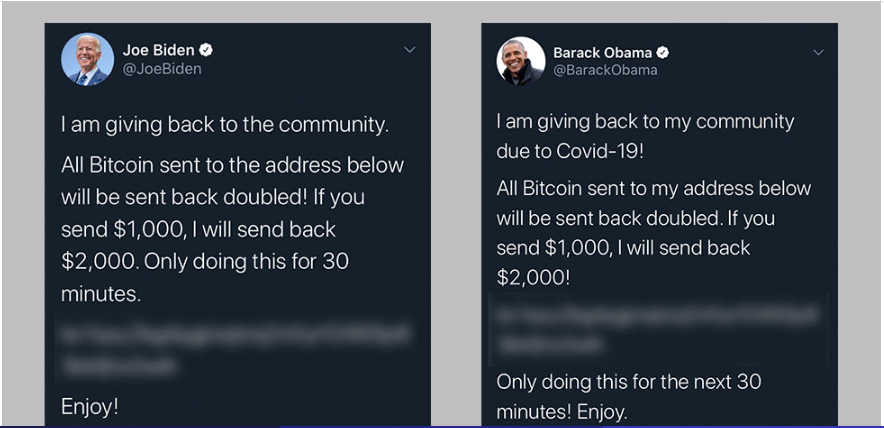
letters@hindustan

Cyber attacks since COVID outbreak (Contd.)

Cyber criminals sincere in lockdown

Twitter Hack

- 130 high-profile twitter accounts were hacked (last week)
- $**120K** worth bitcoins were lost



Joe Biden
@JoeBiden

I am giving back to the community.

All Bitcoin sent to the address below will be sent back doubled! If you send $1,000, I will send back $2,000. Only doing this for 30 minutes.

[redacted]

Enjoy!

Barack Obama
@BarackObama

I am giving back to my community due to Covid-19!

All Bitcoin sent to my address below will be sent back doubled. If you send $1,000, I will send back $2,000!

[redacted]

Only doing this for the next 30 minutes! Enjoy.

Thank You for Your Attention!