

INDEX

Page numbers in italics refer to figures and tables.

A

ACLs (access control lists), 13, 46, 66, 71, 74
Active Directory, querying for users and groups, 123–127
Akamai, 12, 66
alert fatigue, 23, 33
Amazon EC2. *See* EC2
Amazon EventBridge, 212, 216–217, 217
Amazon VPC. *See* VPCs
Amazon Web Services. *See* AWS
API Gateway, 13, 203
application design and development
 principles, 3–19
 loggers, 10–11
 multistage builds, 16–19
 naming standards, 10, 10
 redirection infrastructure, 11–12
 runtime requirements, 13–15
 securing infrastructure, 12–13
 syntax conventions, 10
 three Rs, 4–10
 reliability, 8–10
 reusability, 6–8
 robustness, 4–6
 vs. tool modification, 15–16
authentication
 client-side, 13
 CSP services, 66
 multifactor, 12
 SSH, 98–99, 113–114
AWS (Amazon Web Services)
 accounts, 38, 228, 230
 enumeration with traffic redirection, 38–40, 39
 permissions, 228–229, 229
 polling commands from service queues, 7

 redirection infrastructure, 12
 resource hierarchy, 228
 root user, 229
 securing infrastructure, 13
 VPC networking, 229
AWS API Gateway, 13, 203
AWS CloudFormation. *See* CloudFormation
AWS CloudTrail. *See* CloudTrail
AWS CloudWatch. *See* CloudWatch
AWS Lambda. *See* Lambda
AWS Management Console. *See* Management Console
AWS Systems Manager (SSM), 40, 50

B

behavioral indicators of compromise (BIOCs), 118
BigQuery, 67
binaries. *See also* LOLBins
 avoiding risky, 28–30
 binary structure of C and Go, 31–32, 32
 importing precompiled, 112
binary analyzer
 building/running directly, 186
 enumerating compiled binary headers, 183–187
 extending to return exit codes, 187
 output, 186, 187
Binject, 141
BIOCs (behavioral indicators of compromise), 118

C

C2 (command-and-control) builds, 65–96
 advanced functionality, 93–96
 deployment architecture, 66–67, 67
 execution flow, 5–6, 6.

- C2 builds (*continued*)
 - prototype creation, 68–82
 - base implant code, 76–82
 - Go modules and dependencies, 75–76
 - preparing GCP environment, 68–75
 - redirection infrastructure, 12, 12
 - reliability and exception handling, 8–9, 9
 - reusability and modular coding, 6–7
 - robustness and requirements
 - gathering, 4–6
 - technical requirements, 66
 - usability and stealth improvement, 82–93
 - configuring Firestore permissions, 83–84
 - creating Firestore database, 90–91
 - embedding credentials and adding Firestore, 84–90
 - testing and execution, 91–93
- CAB files
 - exfiltration client, 160–161, 161, 169–172
 - staging tool, 160–161, 161, 164–169, 168
- calc_triangle_area() function, 7
- CDN (content delivery network)
 - providers, 12–13, 66
- CFData construct, 164
- CFFolder construct, 164
- CFHeader construct, 164
- CI/CD (continuous integration and continuous delivery) pipelines, 22, 187
- CIDR (class interdomain routing), 13, 212
- C language
 - advantages and trade-offs of, 14
 - binary structure, 31–32, 32
 - engineering and forensic tools, 31
- Cloudflare, 12, 66
- CloudFormation (CFN)
 - controlled reveals
 - automating deployment, 206–212
 - deployment architecture, 202, 202
 - simulation stack setup and execution, 206–211, 213, 214, 215
 - defined, 229
 - enumeration with traffic redirection, 55–61
 - reusable deployment templates, 46–50
 - reviewing parameters, 56, 57
 - serverless scanner
 - implementation, 50–55
 - stack creation, 56, 56
 - stack status, 58, 59
 - uploading template files, 56, 56
- example template, 229–230
- Cloud Logging, 159
 - chunked entries, 176–177, 177, 178
 - exfiltration client, 160, 169–172
 - file retrieval client, 175
- cloud persistence, simulating with Lambda IAM, 203–204
- Cloud Security Posture Management (CSPM), 216
- cloud service providers. *See* CSPs
- CloudSQL, 67
- CloudTrail, 202
 - deployment automation, 206
 - detection rule creation, 216–219, 217, 218
 - Lambda IAM operations, 203
 - setting up, 205, 205
 - simulation stack setup and execution, 214, 214
- CloudWatch
 - controlled reveals, 202–204
 - deployment automation, 206–212
 - detection rule creation, 216–219, 217, 218
 - simulation stack setup and execution, 215, 215
 - enumeration with traffic redirection, 40, 42, 63
- C# .NET, 14
- code signing, 31
- command-and-control builds. *See* C2 builds
- compressData() function, 148

- compression
 - data exfiltration, 159–160
 - hybrid packing, 138, 141–143, 148–150, 155–156
 - constants, naming standards for, *10*
 - content delivery network (CDN)
 - providers, 12–13, 66
 - continuous integration and continuous delivery (CI/CD) pipelines, 22, 187
 - controlled reveals, 201–219
 - AWS CloudTrail setup, 205, *205*
 - cloud persistence simulation, 203
 - deployment architecture, 202, *202*
 - deployment automation, 206–212
 - detection rule creation, 216–219, *217, 218*
 - ransomware TTP replication, 204
 - simulation stack setup and execution, 212–216, *213, 214, 215*
 - technical requirements, 202
 - Cortex XDR, 131
 - CrowdStrike, 15
 - CSPM (Cloud Security Posture Management), 216
 - CSPs (cloud service providers)
 - C2 implants, 66
 - enumeration with traffic redirection, 38, 40
 - redirection infrastructure, 12
- D**
- DaC (Detection as Code), 22
 - Datadog, 203, 218
 - data exfiltration, 159–178
 - deployment architecture, 161–162, *161, 162*
 - executing full pipeline, 175–178, *176, 177, 178*
 - exfiltration client, 160–161, *161, 169–172, 175, 176*
 - GCP environment setup, 162–164, *163, 164*
 - retrieval client, 160, 162, *162, 172–175, 176, 176*
 - staging tool, 160–161, 164–169, *168*
 - technical requirements, 161
 - data loss prevention (DLP), 28, 143, 160
 - decodeFile() function, 148–149
 - decompressData() function, 149, 155–156
 - Delphi, 31
 - deobfuscateFromIPv4() function, 149, 155–156
 - Detection as Code (DaC), 22
 - detection checks. *See* controlled reveals
 - detection engineering teams, 22–23, 183, 221
 - detection rules, 216–219, *217, 218*
 - detection tools, 181–200
 - binary analyzer, 182, 183–188
 - deployment architecture, 183, *183*
 - network traffic analyzer, 182, 188–200
 - beaconing histograms, 198, *199*
 - entropy score, 197–198
 - generating traffic log, 188–191
 - monitoring for suspicious traffic, 192–200
 - output, *198*
 - regularity score, 197–198
 - technical requirements, 182
 - DFIR (digital forensics and incident response) analysts, 26
 - DigiCert, 31
 - DLP (data loss prevention), 28, 143, 160
 - Donut, 140–142
 - domain-specific languages (DSLs), *14*
 - dunders (double underscores), 7
- E**
- EC2
 - controlled reveals, 202
 - deployment automation, 206–212
 - replicating ransomware TTPs, 204
 - simulation stack setup and execution, 215–216, *215*
 - defined, 229
 - enumeration with traffic redirection, 39–40
 - Amazon VPC, 45–46
 - deployment, 59, 61
 - detection rule creation, 217–218
 - reusable deployment template, 46–50
 - switching source IP addresses, 63–64
 - user data feature, 212

- EDR (endpoint detection and response)
 - binary analyzer, 183
 - C2 implants, 82, 93
 - cloud-enabled or cloud-managed solutions, 15
 - code signing, 31
 - data exfiltration, 160
 - detection engineering patterns, 22–23
 - entropy levels, 25
 - file padding, 30
 - hybrid packing, 156, 157
 - lateral exploits with worms, 111
 - mimicking end user behavior, 33
 - naming standards, 30
 - resource estimation, 23
 - risky binaries and packages, 28–30
 - switching languages and architecture, 31–32
 - timing optimization, 24
 - transferring executables into shellcode, 139–140
 - ELF (Linkable Format) files, 17–19, 182
 - Encode
 - defined, 138
 - hiding payloads, 142–150
 - encodeFile() function, 148
 - encryption
 - data exfiltration, 160
 - low-entropy, 25–28
 - multistage builds, 16
 - simulating, 204
 - endpoint detection and response. *See* EDR
 - end user behavior, mimicking, 33
 - enumeration
 - without LOLBins, 117–136
 - reasons to avoid LOLBins, 118–119
 - replacing with custom tools, 121–136
 - top categories of, 119–121
 - hostname, 120
 - network and firewall configuration, 120, 120
 - OS, 119
 - running processes, services, and scheduled tasks, 120, 121
 - users and groups, 119, 119
 - with traffic redirection, 37–64
 - deploying AWS Lambda solution, 61–63
 - deploying CloudFormation solution, 55–61
 - deployment architecture, 38–40, 39
 - persistent scanning infrastructure, 45–55
 - switching source IP addresses, 63–64
 - tearing down scanning environment, 64
 - technical requirements, 38
 - web scanner, 40–45
 - epics, 4
 - evasion strategies, 21–34
 - blending into target environment, 25–31
 - avoiding risky binaries and packages, 28–30
 - code signing, 31
 - file padding, 30–31
 - low-entropy encryption, 25–28
 - naming conventions, 30
 - detection engineering patterns, 22
 - false flags, 33
 - mimicking end user behavior, 33
 - resource estimation, 23–24
 - SOC patterns, 22–23
 - switching languages and architecture, 31–32
 - tandem operations, 33
 - timing optimization, 24–25
 - EventBridge, 212, 216–217, 217
 - Ex2Shell
 - converting executables into shellcode, 139–142
 - defined, 138
 - IPv4 obfuscation, 143
 - exception handling
 - Python vs. Go, 223
 - reliability, 8–10
 - exec.Command() function, 133
 - executeShellcode() function, 155–156
 - exfiltration. *See* data exfiltration
- ## F
- f (force overwrite) flag, 118
 - false flags, using for misdirection, 33–34

- Fastly, 66
 - file padding, 30–31
 - file retrieval client, 160, 162, *162*, 172–176, *176*
 - Firestore, 66–67, *67*, 84–90
 - adding documents, 92, *92*
 - creating database, 90–91, *90*
 - embedding credentials, 84–90
 - hierarchical structure, 91
 - permission configuration, 83–84, *83*
 - viewing collections, 91, *91*
 - firewalls
 - common LOLBins, *120*
 - enumerating configuration, 120
 - next-generation capabilities, 12
 - returning configuration details, 130–135
 - Linux, 133–134
 - macOS, 131–133
 - Windows, 130–131
 - web application, 23, 38
 - force overwrite (-f) flag, 118
 - FQDNs (fully qualified domain names), 12
 - functions
 - main, 7
 - naming standards, *10*
 - Step Functions, 50–55, *62*, *63*
 - web response capture, 42–45
- G**
- GCP (Google Cloud Platform). *See also*
 - Cloud Logging; Firestore
 - C2 implants, 66–67
 - basic client, 76–80
 - buckets, 74–75, *74*, *75*, 81–82, *81*
 - preparing environment, 68–75
 - project access, 73–75, *73*, *75*
 - roles, 69–70, *69*, *70*
 - service account, 71–72, *71*, *72*
 - usability and stealth improvement, 82–84
 - data exfiltration, 159
 - chunked entries, 176–177, *177*, *178*
 - creating JSON credentials, 162
 - exfiltration client, 160, 169–172
 - file retrieval client, 175
 - preparing environment, 162–164
 - IAM system roles and permissions, 227, 227
 - Jinja, 50
 - lateral exploit worms, 98–100
 - code, 106–110
 - configuration, 99–100, *100*
 - deploying test infrastructure, 104–105, *104*, *105*
 - SSH password authentication, 113
 - lists of projects, 226, 226
 - project setup, 228
 - redirection infrastructure, 12
 - resource hierarchy, 226
 - service account keys, 227–228, 228
 - getCredentialJSON() function, 84
 - Ghidra, 31
 - Go
 - advantages and trade-offs of, *14*
 - binary analyzer, 183–187
 - binary structure, 31–32, *32*
 - C2 implants
 - base code, 80
 - creating limited custom LOLBin replacements, 93–95, *95*
 - Firestore permissions, 83
 - module and dependency setup, 75–76
 - common troubleshooting and operation commands, 225–226
 - data exfiltration, 164
 - downloading, 224
 - enumeration without LOLBins, 117–118, 121
 - firewall configuration details, 130–134
 - hostname extraction and display, 127–128
 - identifying running processes, services, and scheduled tasks, 135–136
 - network configuration details, 128–130
 - OS detection and reporting, 122
 - querying AD for users and groups, 123–126
 - error handling, 223
 - “hello world” program, 224
 - hybrid packing, 140–141, 150

Go (*continued*)
 lateral exploit worms, 98, 106–110,
 112–113
 linter, 224
 module directory, 225
 setting up, 224
 switching languages, 31
 triggering compiler errors, 222–223
 variable declaration, 222
 go-donut library, 140
 GoLogExfil, 160–161, *161*, 169–172,
 175, *176*
 GoLogRetrieve, 162, *162*, 176, *176*
 Google Cloud Platform. *See* GCP
 GoReSym, 31
 GoStageCab, 160–161, *161*, 164–169, *168*
 GTFOBins, 28

H

HashiCorp Terraform. *See* Terraform
 hostname
 enumerating, 120
 extraction and display, 127–128
 hostname command, 117, 120
 hybrid packing, 137–157
 executables to shellcode generation,
 139–142
 hiding payload, 142–150
 in-memory payload execution,
 150–157, *157*
 tooling and processes, 138, *139*

I

IAM (Identity and Access
 Management)
 AWS, 228–229
 C2 implants, 68, *69*, 71, 73–76, 83–84
 controlled reveals, 205–206,
 212–214, 216
 cloud persistence simulation, 203
 data exfiltration, 163, *164*
 enumeration with traffic redirection,
 50, 55, 58, 58
 GCP, 227
 Identity, Governance, and Administration
 (IGA) processes, 203
 indicators of attack (IOA), 11
 indicators of compromise (IOC), 11–12

in-memory payload execution, 150–157
 interquartile range (IQR), 200
 intrusion prevention systems
 code signing, 31
 detection engineering patterns, 22–23
 network behavior analysis, 182
 resource estimation, 23
 IP addresses
 IPv4 obfuscation, 143–148
 switching source addresses, 63–64
 WAN address checking, 41

J

Jinja, 50

L

Lambda
 client-side authentication, 13
 controlled reveals
 deployment automation, 206–211
 IAM user creation simulation, 203
 simulation stack setup and
 execution, 206–211, *213*, *214*
 defined, 229
 enumeration with traffic redirection
 deployment, 61–63, *62*
 serverless scanner
 implementation, 50–55
 WAN IP address checker, 41
 web scanner construction, 40–45
 lambda_handler() function, 41, 44
 lateral exploit worms, 97–115
 code preparation, 106–111
 compiling and encoding code, 111–112
 deployment architecture, 98, *99*
 infecting first host, 114–115, *115*
 infection instances, 99–106
 deploying test infrastructure,
 100–106
 GCP Cloud Shell and Terraform,
 99–100, *100*
 setting up infection, 112–114
 importing precompiled
 binaries, 112
 manual insertion, 112–113, *113*
 SSH password authentication,
 113–114
 technical requirements, 98

- ldap package, 123
- LDAP server, 127
- Linkable Format (ELF) files, 17–19, 182
- Loader
 - defined, 138
 - in-memory payload execution, 150–157
- loadFromFile() function, 149, 156
- logging and loggers, 182. *See also*
 - Cloud Logging
 - function-level logging, 10–11
 - network traffic analyzer, 188–192
- LOLBins (living-off-the-land binaries), 117–136
 - avoiding, 28, 118–119
 - defined, 13, 118
 - replacing with custom tools, 121–136
 - hostname extraction and display, 127–128
 - network and firewall
 - configuration details, 128–135
 - OS detection and reporting, 122–123
 - querying AD for users and groups, 123–127
 - running processes, services, and scheduled tasks, 135–136
- low-entropy encryption, 25–28
 - decryption routine, 26
 - hex support, 26–28
 - working example, 25–26

M

- main functions, purpose of, 7
- Management Console
 - controlled reveals
 - CloudTrail setup, 205, 205
 - detection rule creation, 216, 218–219
 - simulation stack setup and execution, 212–216
 - documentation, 45
 - enumeration with traffic redirection
 - deployment, 55, 61, 62
 - Lambda setup, 40
 - web response capture, 44–45
- Mandiant GoReSym, 31

- Metasploit, 15
- Microsoft cabinet files. *See* CAB files
- Microsoft Defender, 15, 24
- Mimikatz, 15
- MITRE ATT&CK, 118
- modular coding, 6–8
- multifactor authentication (MFA), 12
- multistage builds, 16–19, 17

N

- naming standards
 - adopting, 10, 10
 - blending into target environment, 30
- network address translation (NAT)
 - ACLs, 13
 - NAT Gateway, 45
- network configuration
 - common LOLBins, 120
 - enumerating, 120
 - returning details of, 128–130
- network traffic analyzer, 188–200
 - beaconing histograms, 198, 199
 - entropy score, 197–198
 - monitoring for suspicious traffic, 192–200
 - output, 198
 - regularity score, 197–198
 - traffic log, 188–191
- Nishang, 15

O

- obfuscateAsIPv4() function, 148
- obfuscation
 - in-memory payload, 142–150
 - IPv4, 143–148
- OIDC (OpenID Connect), 228
- operating system (OS)
 - common LOLBins, 119
 - detection and reporting, 122–123
 - enumeration, 119
- os.Hostname() function, 127
- os.Open() function, 127
- os.Stat() function, 110

P

- packages, risky, 28–30
- packers
 - defined, 138

- packers (*continued*)
 - hybrid packing, 137–157
 - executables to shellcode generation, 139–142
 - hiding payload, 142–150
 - in-memory payload execution, 150–157
 - tooling and processes, 138
 - workflow, 139
 - multistage builds, 16–18, 17
 - Palo Alto Networks
 - Cortex XDR, 188–192
 - firewall traffic log, 188–191
 - password authentication, 99, 113–114
 - password spraying, 98, 111
 - payload obfuscation, 142–150
 - permissions
 - AWS, 228–229, 229
 - Firestore, 83–84, 83
 - GCP, 227, 227
 - persistent scanning infrastructure, 45–55
 - Amazon VPC, 45–46
 - CloudFormation deployment template, 46–50
 - Step Functions, 50–55
 - PowerShell, 12, 13, 27, 141, 175
 - precompiled binaries, importing, 112
 - primitives, 68
 - processes, running
 - common LOLBins, 121
 - enumerating, 120–121
 - identifying, 135–136
 - Python
 - advantages and trade-offs of, 14
 - dunders, 7
 - function-level logging, 10–11
 - network traffic analyzer, 188–200
 - polling commands from AWS service queues, 7
 - web response capture, 42–45
- R**
- random number generation, 24
 - RansomHub, 218
 - ransomware TTPs, replicating, 204
 - Red Canary, 118
 - redirection
 - defined, 11
 - designing infrastructure, 11–12
 - enumeration with traffic redirection, 37–64
 - reliability, 8–10
 - requirements gathering, 4–6
 - matrices, 5, 5
 - resource estimation, 23–24
 - retrieval client, 160, 162, 162, 172–176, 176
 - return on investment (ROI), 5
 - reusability
 - defined, 6
 - modular coding, 6–8
 - robustness
 - defined, 4
 - requirements gathering, 4–6
 - runCommand() function, 95
 - runtime requirements, 13–15, 14
- S**
- saveToFile() function, 148
 - scheduled tasks, running
 - common LOLBins, 121
 - enumerating, 120–121
 - identifying, 135–136
 - Secure Access Service Edge (SASE), 45
 - secure copy protocol (SCP), 111
 - securing infrastructure, 12–13
 - Security Operations Center teams.
 - See* SOC teams
 - self_delete() function, 9
 - services, running
 - common LOLBins, 121
 - enumerating, 120–121
 - identifying, 135–136
 - Shannon Entropy Scale, 25
 - shellcode, converting executables to, 139–142
 - ShellcodeFromBytes() function, 141–142
 - SIEM (security information and event management)
 - detection engineering patterns, 22–23
 - detection rule creation, 216
 - entropy levels, 25
 - network behavior analysis, 182–183

- resource estimation, 23
- timing optimization, 24
- site reliability engineering (SRE), 218
- SOAR, 183
- SOC (Security Operations Center) teams
 - alert fatigue, 23, 33
 - avoiding LOLBins, 118
 - patterns, 22–23
 - resource estimation, 23
- source IP addresses, switching, 63–64
- Splunk, 218
- SSH password authentication, 99
 - enabling, 113–114
- Stackdriver, 160
- Step Functions, 50–55, 62, 63
- strings command, 15
- suspicious traffic, monitoring for, 192–200
- syntax conventions, 10
- syntax linters, 10
- Systems Manager (SSM), 40, 50

T

- tactics, techniques, and procedures.
 - See* TTPs
- Tailscale
 - accounts, 38
 - enumeration with traffic redirection, 38–40, 50
 - deployment, 57–61, 57, 58, 59, 60, 61
 - reusable deployment template, 46–50
 - shutting down, 64
 - switching source IP addresses, 63
- tandem operations, 33
- tcpdump, 11, 81
- Terraform
 - IaC, 50, 229
 - lateral exploit worms
 - configuration, 99–100
 - deploying test infrastructure, 100–104, 105
- terraform plan command, 104, 105
- Thawte, 31
- three Rs of application development, 4–10
 - reliability, 8–10

- reusability, 6–8
- robustness, 4–6
- timing optimization, 24–25
- TLS encryption
 - C2 builds, 4, 66
 - loggers, 10
 - multistage builds, 16, 19
 - redirection infrastructure, 12
 - web response capture, 44
- tool modification vs. development, 15–16
- traffic logging, 182, 188–191. *See also* Cloud Logging
- traffic redirection, enumeration with, 37–64
 - deploying AWS Lambda solution, 61–63
 - deploying CloudFormation solution, 55–61
 - deployment architecture, 38–40, 39
 - persistent scanning infrastructure, 45–55
 - Amazon VPC, 45–46
 - reusable deployment template, 46–50
 - Step Functions, 50–55
 - switching source IP addresses, 63–64
- tearing down scanning environment, 64
- technical requirements, 38
- web scanner, 40–45
 - AWS Lambda, 40–42
 - Python function, 42–45
- Transport Layer Security. *See* TLS encryption
- TTPs (tactics, techniques, and procedures). *See also* network traffic analyzer
- false flags, 33
- prioritization, 187
- replicating ransomware on EC2, 204
- tandem operations, 33
- time optimization, 24

U

- urlcache feature, 118
- user behavior, mimicking, 33

- users and groups
 - common LOLBins, 119
 - enumeration, 119–120
 - querying Active Directory for, 123–127
- user stories, 4–5

V

- validation tests. *See* controlled reveals
- variables, naming standards for, 10
- Varonis, 218
- Verisign, 31
- VirusTotal, 12
- VPCs (virtual private clouds), 228–229
 - controlled reveals, 202, 206–212
 - enumeration with traffic redirection, 39–40
 - persistent scanning infrastructure, 45–46
 - reusable deployment template, 46–49
 - web scanner construction, 40
- lateral exploit worms, 99–103

W

- WAFs (web application firewalls), 23, 38
 - AWS Lambda, 40–42
 - persistent scanning infrastructure, 45–55
 - Amazon VPC, 45–46
 - reusable deployment template, 46–50
 - Step Functions, 50–55
 - Python function for capturing web responses, 42–45
- Webhook Test, 12
- web scanners, 40–45
- whoami, 93, 117
- Windows Defender, 131
- Windows Sysmon, 22
- Wireshark, 81
- worms. *See* lateral exploit worms