Extracting Threat Intelligence From Cheat Binaries For Anti-Cheating

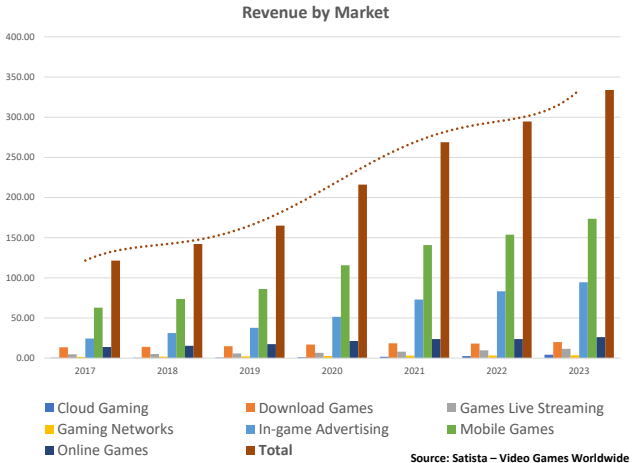
Md Sakib Anwar Chaoshun Zuo Carter Yagemann Zhiqiang Lin

RAID 2023

Current State of Gaming

Cheating is the biggest threat

1. **37%** of gamers have confessed to cheating
2. **77%** of gamers may stop playing the game once exposed to cheating

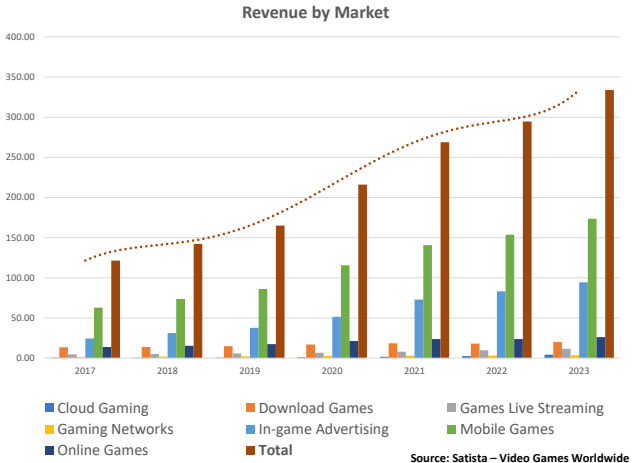


Current State of Gaming

Cheating is the biggest threat

1. **37%** of gamers have confessed to cheating
2. **77%** of gamers may stop playing the game once exposed to cheating

How do cheating continue to exist against today's security?



Cheating in 2023

Bob

Anti Cheat

Obfuscation

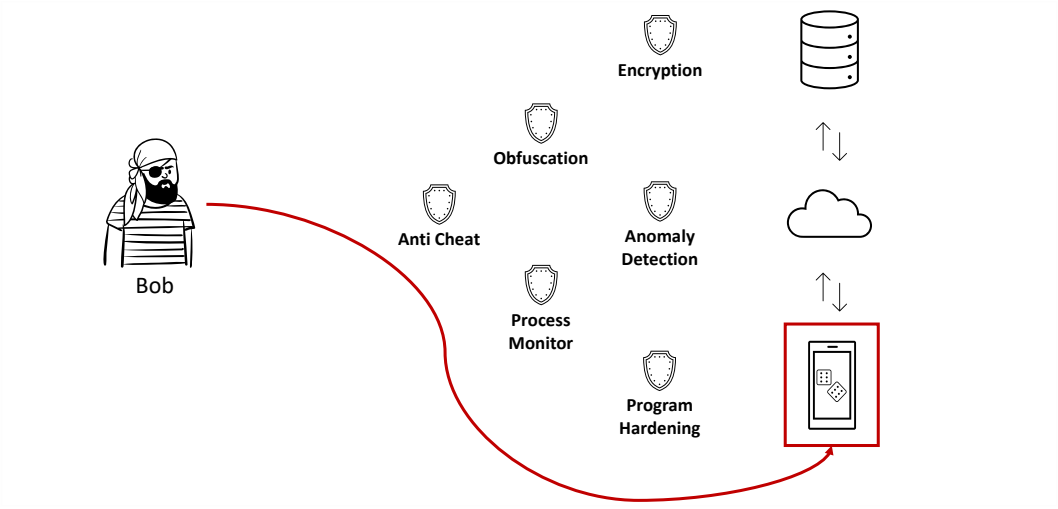
Process Monitor

Encryption

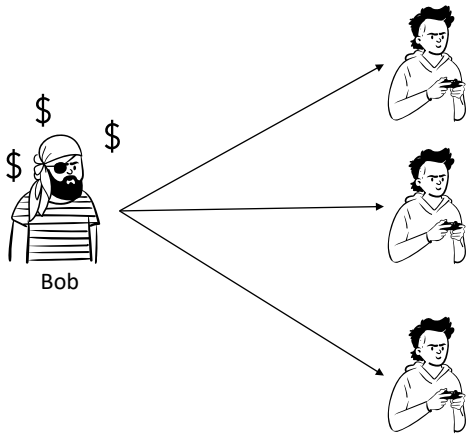
Anomaly Detection

Program Hardening

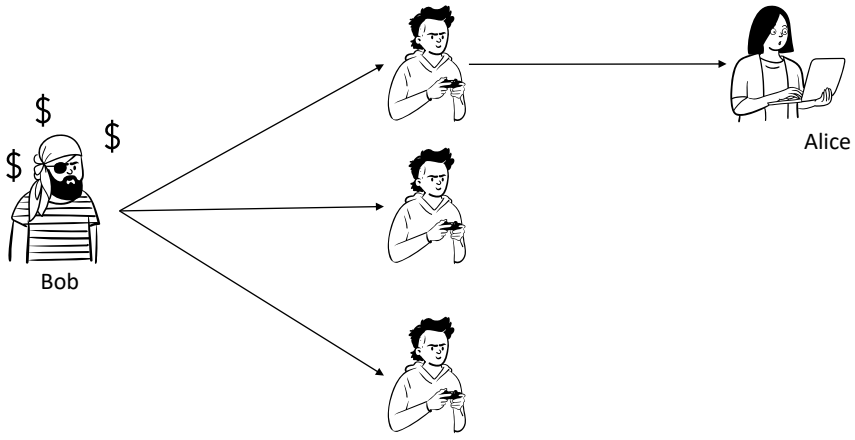
Cheating in 2023



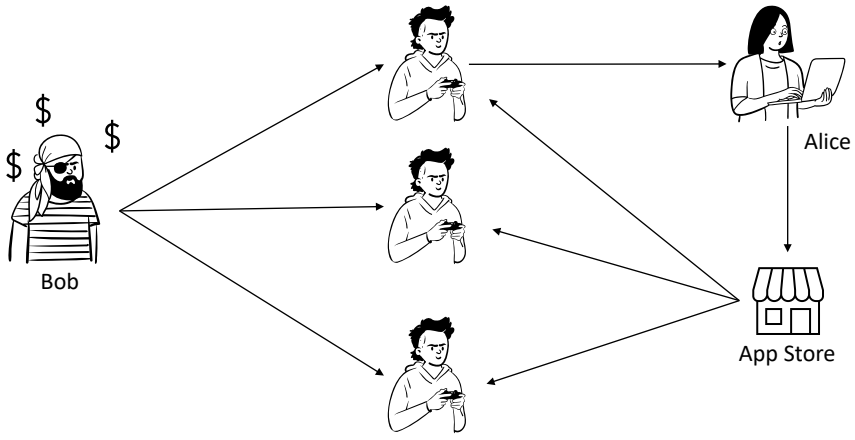
Defense, Loophole & Proposition



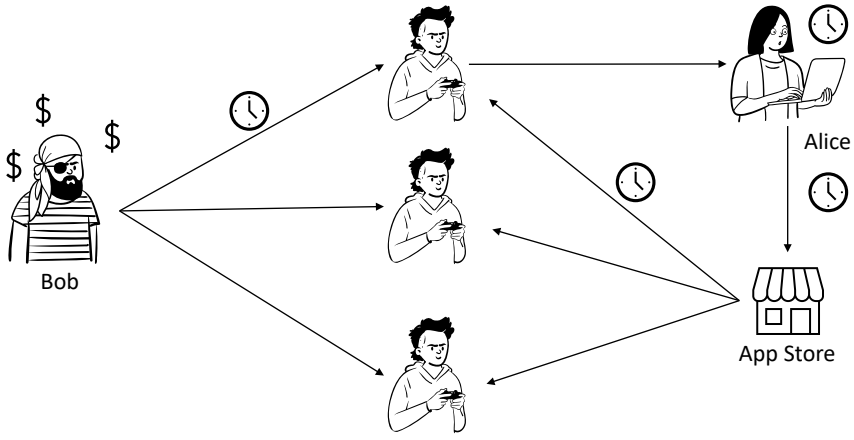
Defense, Loophole & Proposition



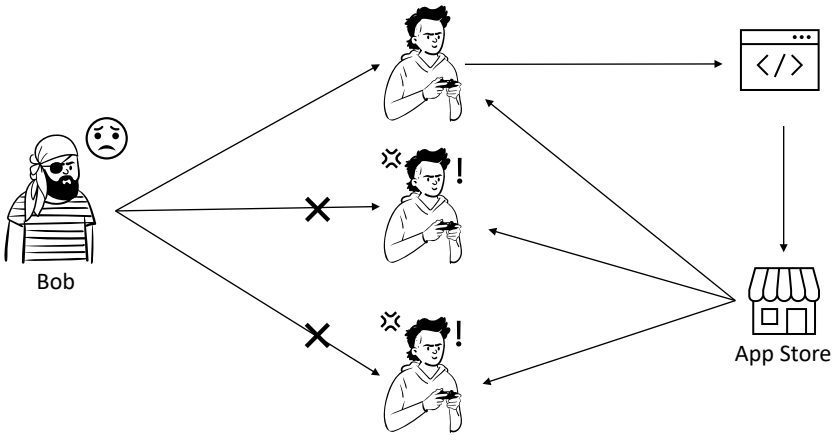
Defense, Loophole & Proposition



Defense, Loophole & Proposition

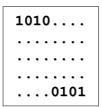


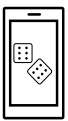
Defense, Loophole & Proposition



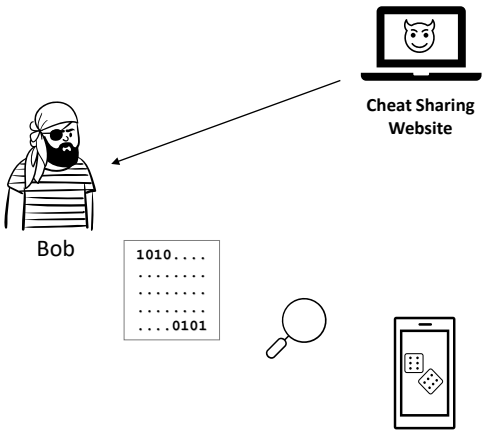
Cheat Making

Bob



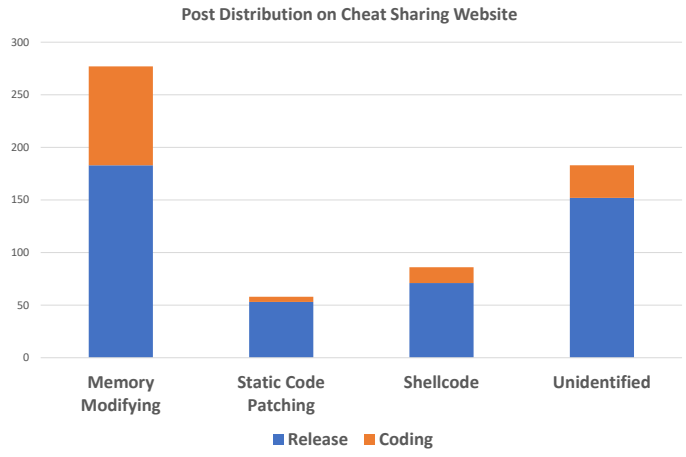


Cheat Making

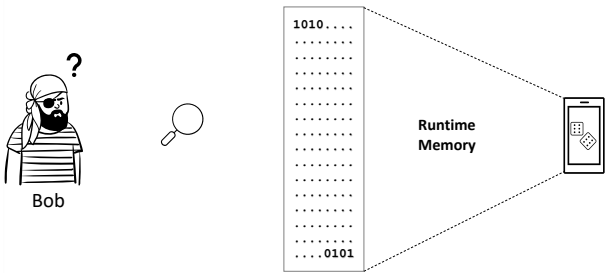


Cheat Making

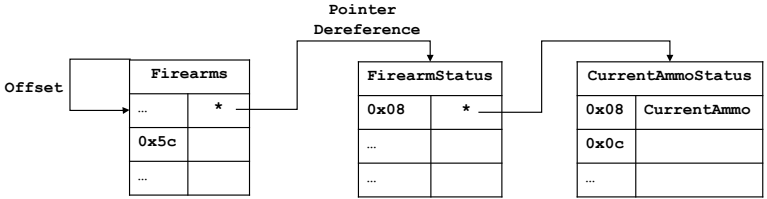

Bob



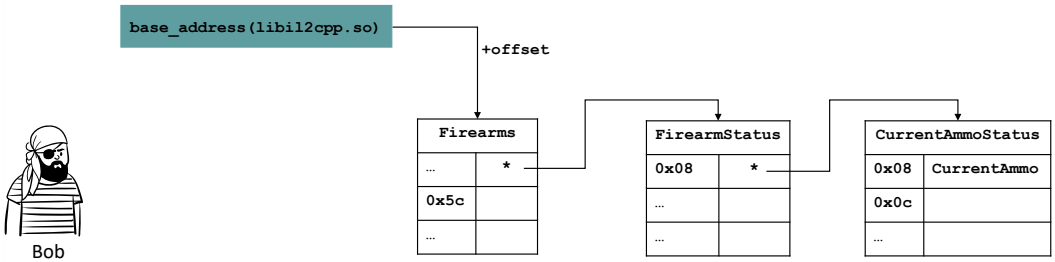
Memory Modifying Cheat



Memory Modifying Cheat



Memory Modifying Cheat



Threat Intelligence: Memory Access Graph (MAG)

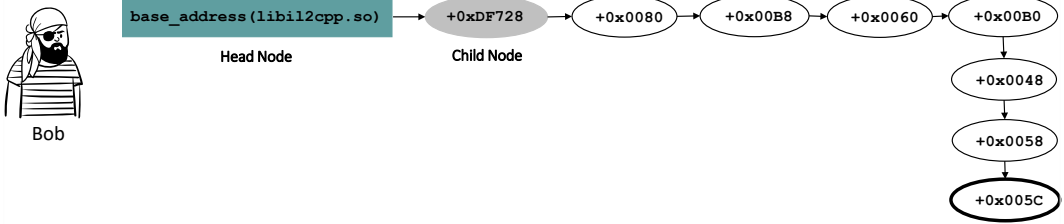


Figure: MAG in Real World Cheat for San Andreas Unity

How to Extract MAG for Automated Defense?

```
int main(int argc, char** argv) {
    ..
    memfd = fopen("/proc/self/mem", "r");
    library_name = "libil2cpp.so";
    base_address = get_module_base((char *) &library_name);
    add = base_address + 0x820cc24;
    lVar1 = readValueL(add);
    add2 = lVar1 + 0x50;
    lVar1 = readValueL(add2);
    add3 = lVar1 + 0x8;
    ..
}
ulong get_module_base((char *) &library_name) {
    ..
    FILE *mapping = fopen("/proc/self/maps", "r");
    ..
    char *pcVar1, *base_address_string;
    ulong base_address; char file_content[1024];
    do{
        pcVar1 = fgets(file_content,0x400,local_18);
        ..
        pcVar1 = strstr(file_content, library_name);
    } while (pcVar1 == (char *)0x0);
    base_address_string = strtok(file_content,"-");
    base_address = strtoul(base_address_string,
        (char **)0x0,0x10);
    return base_address;
}
undefined8 readValueL(__off64_t param_1) {
    ..
    pread64(memfd,&local_8,0x4,param_1);
    return local_8;
}
```

Step 1: Locating Pointer Chain Head with API Signature

```
1  int main(int argc, char** argv) {
2      ..
3      memfd = fopen("/proc/self/mem", "r");
4      library_name = "libil2cpp.so";
5      base_address = get_module_base((char *) &library_name);
6      add = base_address + 0x820cc24;
7      lVar1 = readValueL(add);
8      add2 = lVar1 + 0x50;
9      lVar1 = readValueL(add2);
10     add3 = lVar1 + 0x8;
11     ..
12 }
13 ulong get_module_base((char *) &library_name) {
14     ..
15     FILE *mapping = fopen("/proc/self/maps", "r");
16     ..
17     char *pcVar1, *base_address_string;
18     ulong base_address; char file_content[1024];
19     do{
20         pcVar1 = fgets(file_content,0x400,local_18);
21         ..
22         pcVar1 = strstr(file_content, library_name);
23     } while (pcVar1 == (char *)0x0);
24     base_address_string = strtok(file_content,"-");
25     base_address = strtoul(base_address_string,
           (char **)0x0,0x10);
26     return base_address;
27 }
28 undefined8 readValueL(__off64_t param_1) {
29     ..
30     pread64(memfd,&local_8,0x4,param_1);
31     return local_8;
32 }
```

Step 1: Locating Pointer Chain Head with API Signature

```
1  int main(int argc, char** argv) {
2      ..
3      memfd = fopen("/proc/self/mem", "r");
4      library_name = "libil2cpp.so";
5      base_address = get_module_base((char *) &library_name);
6      add = base_address + 0x820cc24;
7      lVar1 = readValueL(add);
8      add2 = lVar1 + 0x50;
9      lVar1 = readValueL(add2);
10     add3 = lVar1 + 0x8;
11     ..
12 }
13 ulong get_module_base((char *) &library_name) {
14     ..
15     FILE *mapping = fopen("/proc/self/maps", "r");
16     ..
17     char *pcVar1, *base_address_string;
18     ulong base_address; char file_content[1024];
19     do{
20         pcVar1 = fgets(file_content,0x400,local_18);
21         ..
22         pcVar1 = strstr(file_content, library_name);
23     } while (pcVar1 == (char *)0x0);
24     base_address_string = strtok(file_content,"-");
25     base_address = strtoul(base_address_string,
           (char **)0x0,0x10);
26     return base_address;
27 }
28 undefined8 readValueL(__off64_t param_1) {
29     ..
30     pread64(memfd,&local_8,0x4,param_1);
31     return local_8;
32 }
```

Step 1: Locating Pointer Chain Head with API Signature

```
1  int main(int argc, char** argv) {
2      ..
3      memfd = fopen("/proc/self/mem", "r");
4      library_name = "libil2cpp.so";
5      base_address = get_module_base((char *) &library_name);
6      add = base_address + 0x820cc24;
7      lVar1 = readValueL(add);
8      add2 = lVar1 + 0x50;
9      lVar1 = readValueL(add2);
10     add3 = lVar1 + 0x8;
11     ..
12 }
13 ulong get_module_base((char *) &library_name) {
14     ..
15     FILE *mapping = fopen("/proc/self/maps", "r");
16     ..
17     char *pcVar1, *base_address_string;
18     ulong base_address; char file_content[1024];
19     do{
20         pcVar1 = fgets(file_content,0x400,local_18);
21         ..
22         pcVar1 = strstr(file_content, library_name);
23     } while (pcVar1 == (char *)0x0);
24     base_address_string = strtok(file_content,"-");
25     base_address = strtoul(base_address_string,
           (char **)0x0,0x10);
26     return base_address;
27 }
28 undefined8 readValueL(__off64_t param_1) {
29     ..
30     pread64(memfd,&local_8,0x4,param_1);
31     return local_8;
32 }
```

Step 1: Locating Pointer Chain Head with API Signature

```
1  int main(int argc, char** argv) {
2      ..
3      memfd = fopen("/proc/self/mem", "r");
4      library_name = "libil2cpp.so";
5      base_address = get_module_base((char *) &library_name);
6      add = base_address + 0x820cc24;
7      lVar1 = readValueL(add);
8      add2 = lVar1 + 0x50;
9      lVar1 = readValueL(add2);
10     add3 = lVar1 + 0x8;
11     ..
12 }
13 ulong get_module_base((char *) &library_name) {
14     ..
15     FILE *mapping = fopen("/proc/self/maps", "r");
16     ..
17     char *pcVar1, *base_address_string;
18     ulong base_address; char file_content[1024];
19     do{
20         pcVar1 = fgets(file_content,0x400,local_18);
21         ..
22         pcVar1 = strstr(file_content, library_name);
23     } while (pcVar1 == (char *)0x0);
24     base_address_string = strtok(file_content,"-");
25     base_address = strtoul(base_address_string,
           (char **)0x0,0x10);
26     return base_address;
27 }
28 undefined8 readValueL(__off64_t param_1) {
29     ..
30     pread64(memfd,&local_8,0x4,param_1);
31     return local_8;
32 }
```

Step 1: Locating Pointer Chain Head with API Signature

```
1  int main(int argc, char** argv) {
2      ..
3      memfd = fopen("/proc/self/mem", "r");
4      library_name = "libil2cpp.so";
5      base_address = get_module_base((char *) &library_name);
6      add = base_address + 0x820cc24;
7      lVar1 = readValueL(add);
8      add2 = lVar1 + 0x50;
9      lVar1 = readValueL(add2);
10     add3 = lVar1 + 0x8;
11     ..
12 }
13 ulong get_module_base((char *) &library_name) {
14     ..
15     FILE *mapping = fopen("/proc/self/maps", "r");
16     ..
17     char *pcVar1, *base_address_string;
18     ulong base_address; char file_content[1024];
19     do{
20         pcVar1 = fgets(file_content,0x400,local_18);
21         ..
22         pcVar1 = strstr(file_content, library_name);
23     } while (pcVar1 == (char *)0x0);
24     base_address_string = strtok(file_content,"-");
25     base_address = strtoul(base_address_string,
           (char **)0x0,0x10);
26     return base_address;
27 }
28 undefined8 readValueL(__off64_t param_1) {
29     ..
30     pread64(memfd,&local_8,0x4,param_1);
31     return local_8;
32 }
```

Step 2: Locating Valid Offset & Memory Access with DDG

```
1  int main(int argc, char** argv) {
2      ..
3      memfd = fopen("/proc/self/mem", "r");
4      library_name = "libil2cpp.so";
5      base_address = get_module_base((char *) &library_name);
6      add = base_address + 0x820cc24;
7      lVar1 = readValueL(add);
8      add2 = lVar1 + 0x50;
9      lVar1 = readValueL(add2);
10     add3 = lVar1 + 0x8;
11     ..
12 }
13 ulong get_module_base((char *) &library_name) {
14     ..
15     FILE *mapping = fopen("/proc/self/maps", "r");
16     ..
17     char *pcVar1, *base_address_string;
18     ulong base_address; char file_content[1024];
19     do{
20         pcVar1 = fgets(file_content,0x400,local_18);
21         ..
22         pcVar1 = strstr(file_content, library_name);
23     } while (pcVar1 == (char *)0x0);
24     base_address_string = strtok(file_content,"-");
25     base_address = strtoul(base_address_string,
           (char **)0x0,0x10);
26     return base_address;
27 }
28 undefined8 readValueL(__off64_t param_1) {
29     ..
30     pread64(memfd,&local_8,0x4,param_1);
31     return local_8;
32 }
```

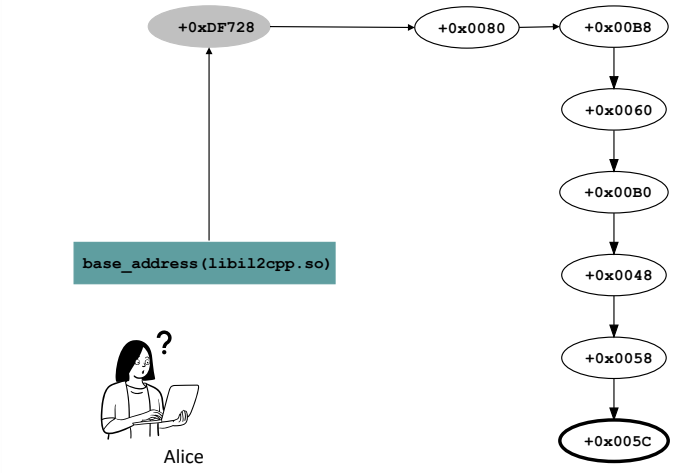
Step 2: Locating Valid Offset & Memory Access with DDG

```
1  int main(int argc, char** argv) {
2      ..
3      memfd = fopen("/proc/self/mem", "r");
4      library_name = "libil2cpp.so";
5      base_address = get_module_base((char *) &library_name);
6      add = base_address + 0x820cc24;
7      lVar1 = readValueL(add);
8      add2 = lVar1 + 0x50;
9      lVar1 = readValueL(add2);
10     add3 = lVar1 + 0x8;
11     ..
12 }
13 ulong get_module_base((char *) &library_name) {
14     ..
15     FILE *mapping = fopen("/proc/self/maps", "r");
16     ..
17     char *pcVar1, *base_address_string;
18     ulong base_address; char file_content[1024];
19     do{
20         pcVar1 = fgets(file_content,0x400,local_18);
21         ..
22         pcVar1 = strstr(file_content, library_name);
23     } while (pcVar1 == (char *)0x0);
24     base_address_string = strtok(file_content,"-");
25     base_address = strtoul(base_address_string,
           (char **)0x0,0x10);
26     return base_address;
27 }
28 undefined8 readValueL(__off64_t param_1) {
29     ..
30     pread64(memfd,&local_8,0x4,param_1);
31     return local_8;
32 }
```

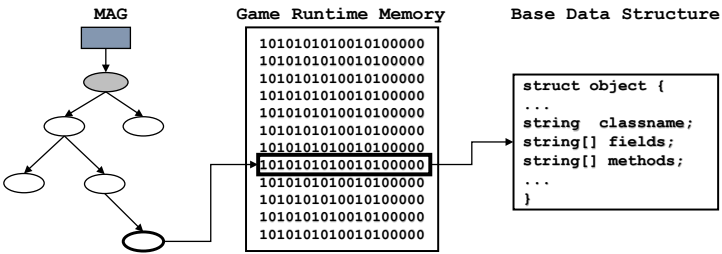
Step 2: Locating Valid Offset & Memory Access with DDG

```
1  int main(int argc, char** argv) {
2      ..
3      memfd = fopen("/proc/self/mem", "r");
4      library_name = "libil2cpp.so";
5      base_address = get_module_base((char *) &library_name);
6      add = base_address + 0x820cc24;
7      lVar1 = readValueL(add);
8      add2 = lVar1 + 0x50;
9      lVar1 = readValueL(add2);
10     add3 = lVar1 + 0x8;
11     ..
12 }
13 ulong get_module_base((char *) &library_name) {
14     ..
15     FILE *mapping = fopen("/proc/self/maps", "r");
16     ..
17     char *pcVar1, *base_address_string;
18     ulong base_address; char file_content[1024];
19     do{
20         pcVar1 = fgets(file_content,0x400,local_18);
21         ..
22         pcVar1 = strstr(file_content, library_name);
23     } while (pcVar1 == (char *)0x0);
24     base_address_string = strtok(file_content,"-");
25     base_address = strtoul(base_address_string,
           (char **)0x0,0x10);
26     return base_address;
27 }
28 undefined8 readValueL(__off64_t param_1) {
29     ..
30     pread64(memfd,&local_8,0x4,param_1);
31     return local_8;
32 }
```

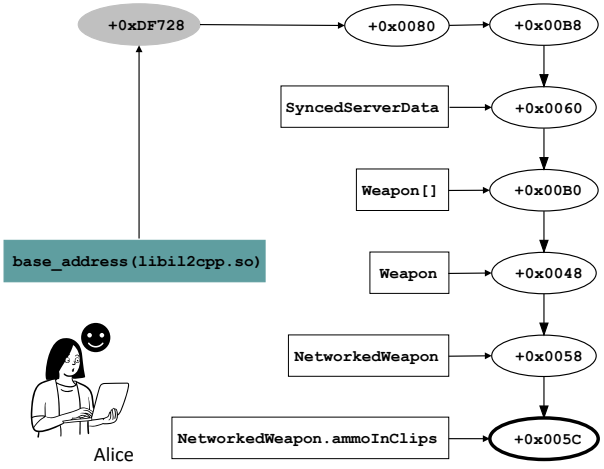
Step 3: Mapping Back to Source with Reflection



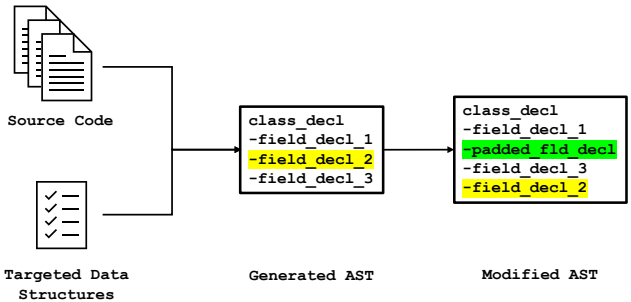
Step 3: Mapping Back to Source with Reflection



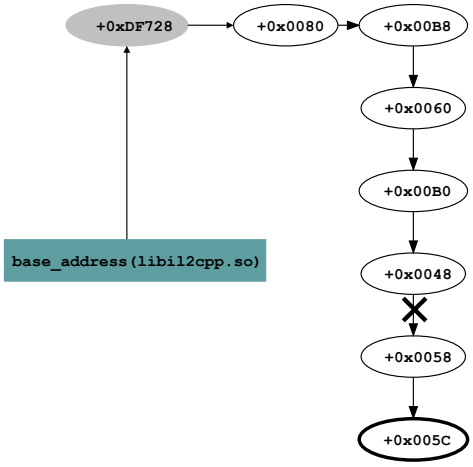
Step 3: Mapping Back to Source with Reflection



Step 4: Automated Defense via Data Structure Randomization

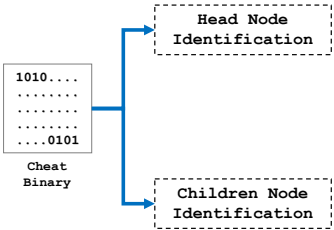


Step 4: Automated Defense via Data Structure Randomization

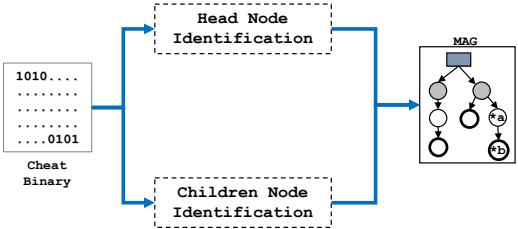


Overall Design

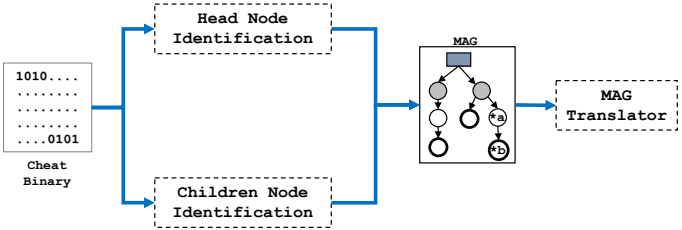
Overall Design



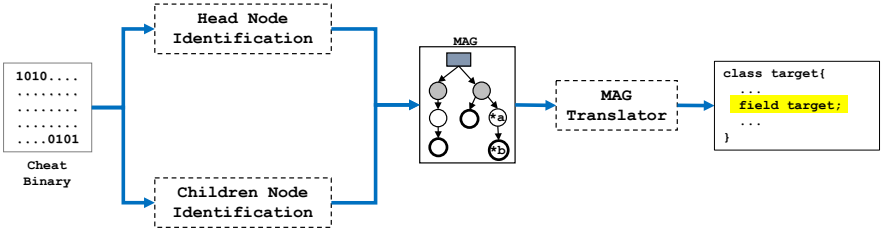
Overall Design



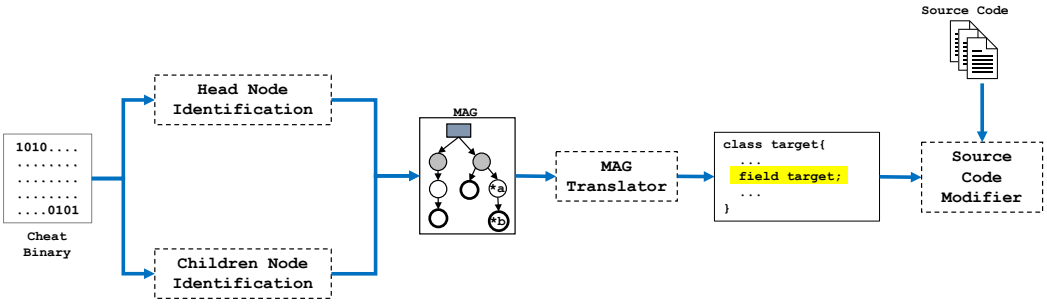
Overall Design



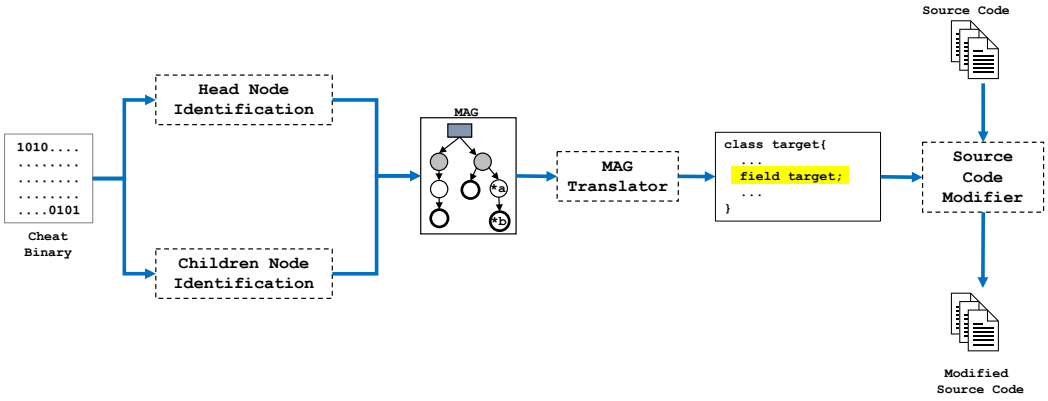
Overall Design



Overall Design



Overall Design



Highlights

Victim Game	Platform	Release	#Installs	Engine	#Cheats	#Binary
CrossFire Mobile	Android	12/3/2015	-	Unity	1	1
Arena 5v5	Android	11/30/2016	10M+	Unity	29	55
PUBG Mobile	Android	3/23/2017	500M+	UE4	35	74
COD Mobile	Android	10/1/2019	100M+	Unity	4	24
Royal Match	Android	2/25/2021	10M+	Unity	1	1
LOL	Android	10/27/2021	-	Unity	1	14
PUBG New State	Android	11/11/2021	10M+	UE4	2	2
Sausage Man	Android	4/29/2022	10M+	Unity	2	18
Assault Cube	Windows	4/1/2022	-	CUBE	1	1
Bard's Tale	Windows	6/17/2005	-	Dark Alliance	1	1
Super Tux	Windows	12/22/2021	-	SuperTux	1	1
COD MW3	Windows	11/8/2011	-	IW	1	1

Highlights

Victim Game	Platform	Release	#Installs	Engine	#Cheats	#Binary
CrossFire Mobile	Android	12/3/2015	-	Unity	1	1
Arena 5v5	Android	11/30/2016	10M+	Unity	29	55
PUBG Mobile	Android	3/23/2017	500M+	UE4	35	74
COD Mobile	Android	10/1/2019	100M+	Unity	4	24
Royal Match	Android	2/25/2021	10M+	Unity	1	1
LOL	Android	10/27/2021	-	Unity	1	14
PUBG New State	Android	11/11/2021	10M+	UE4	2	2
Sausage Man	Android	4/29/2022	10M+	Unity	2	18
Assault Cube	Windows	4/1/2022	-	CUBE	1	1
Bard's Tale	Windows	6/17/2005	-	Dark Alliance	1	1
Super Tux	Windows	12/22/2021	-	SuperTux	1	1
COD MW3	Windows	11/8/2011	-	IW	1	1

Highlights

Victim Game	Platform	Release	#Installs	Engine	#Cheats	#Binary
CrossFire Mobile	Android	12/3/2015	-	Unity	1	1
Arena 5v5	Android	11/30/2016	10M+	Unity	29	55
PUBG Mobile	Android	3/23/2017	500M+	UE4	35	74
COD Mobile	Android	10/1/2019	100M+	Unity	4	24
Royal Match	Android	2/25/2021	10M+	Unity	1	1
LOL	Android	10/27/2021	-	Unity	1	14
PUBG New State	Android	11/11/2021	10M+	UE4	2	2
Sausage Man	Android	4/29/2022	10M+	Unity	2	18
Assault Cube	Windows	4/1/2022	-	CUBE	1	1
Bard's Tale	Windows	6/17/2005	-	Dark Alliance	1	1
Super Tux	Windows	12/22/2021	-	SuperTux	1	1
COD MW3	Windows	11/8/2011	-	IW	1	1

Highlights

Victim Game	Platform	Release	#Installs	Engine	#Cheats	#Binary
CrossFire Mobile	Android	12/3/2015	-	Unity	1	1
Arena 5v5	Android	11/30/2016	10M+	Unity	29	55
PUBG Mobile	Android	3/23/2017	500M+	UE4	35	74
COD Mobile	Android	10/1/2019	100M+	Unity	4	24
Royal Match	Android	2/25/2021	10M+	Unity	1	1
LOL	Android	10/27/2021	-	Unity	1	14
PUBG New State	Android	11/11/2021	10M+	UE4	2	2
Sausage Man	Android	4/29/2022	10M+	Unity	2	18
Assault Cube	Windows	4/1/2022	-	CUBE	1	1
Bard's Tale	Windows	6/17/2005	-	Dark Alliance	1	1
Super Tux	Windows	12/22/2021	-	SuperTux	1	1
COD MW3	Windows	11/8/2011	-	IW	1	1

Highlights

Victim Game	Platform	Release	#Installs	Engine	#Cheats	#Binary
CrossFire Mobile	Android	12/3/2015	-	Unity	1	1
Arena 5v5	Android	11/30/2016	10M+	Unity	29	55
PUBG Mobile	Android	3/23/2017	500M+	UE4	35	74
COD Mobile	Android	10/1/2019	100M+	Unity	4	24
Royal Match	Android	2/25/2021	10M+	Unity	1	1
LOL	Android	10/27/2021	-	Unity	1	14
PUBG New State	Android	11/11/2021	10M+	UE4	2	2
Sausage Man	Android	4/29/2022	10M+	Unity	2	18
Assault Cube	Windows	4/1/2022	-	CUBE	1	1
Bard's Tale	Windows	6/17/2005	-	Dark Alliance	1	1
Super Tux	Windows	12/22/2021	-	SuperTux	1	1
COD MW3	Windows	11/8/2011	-	IW	1	1

Highlights

Victim Game	Platform	Release	#Installs	Engine	#Cheats	#Binary
CrossFire Mobile	Android	12/3/2015	-	Unity	1	1
Arena 5v5	Android	11/30/2016	10M+	Unity	29	55
PUBG Mobile	Android	3/23/2017	500M+	UE4	35	74
COD Mobile	Android	10/1/2019	100M+	Unity	4	24
Royal Match	Android	2/25/2021	10M+	Unity	1	1
LOL	Android	10/27/2021	-	Unity	1	14
PUBG New State	Android	11/11/2021	10M+	UE4	2	2
Sausage Man	Android	4/29/2022	10M+	Unity	2	18
Assault Cube	Windows	4/1/2022	-	CUBE	1	1
Bard's Tale	Windows	6/17/2005	-	Dark Alliance	1	1
Super Tux	Windows	12/22/2021	-	SuperTux	1	1
COD MW3	Windows	11/8/2011	-	IW	1	1

Detailed Result

Platform	Game	Hash	Input #Bin	Input ∑Size	Output Height	Output #Branch	Output #Edges	Output #Base	Output Bases	Memory Access Read	Memory Access Write
Android	CFM	4e349c25d1c5e303f73b9fa8b94934dd	1	652	1	3	3	2	[TS][U]	✓	X
	Arena 5v5	0a45409737c036f9d59c5feb427ba9c5	1	12	1	1	1	1	[GC]	✓	X
		2e8268d32dc22c31dd8579bca6b7f7d7	4	48	1	1	1	2	[GC][U]	✓	X
		* (10)	2	24	6	8	25	1	[GC]	✓	X
		...									
		c632aeaaecbe67487f0bf6f69416cb38	1	21	8	16	55	2	[GC][U]	✓	X
		76bab2ee423c05c1b6f10abc52653683	1	358	10	14	122	2	[GC][IL]	✓	✓
		04caab0a8f0b10d7750ae1d424034a7b	3	41	10	22	74	3	[GC][U][IL]	✓	X
	PUBG	397446459fe284a2c10f676b57c03982	1	13	1	1	1	1	[UE4]	✓	✓
		...									
		9a2cec9ac23cc6b9713d983d202a04ed	1	13	5	1	5	1	[UE4]	✓	✓
	COD	* (3)	1	12	1	1	1	1	[IL]	✓	X
		ac97f45290f238e5346f0ef5ae839cb9	21	269	1	3	3	3	[IL][GC][U]	✓	✓
	Royal Match	c8b4767dc7a0b57ce608173cbc7e6b15	1	8	6	1	7	1	[IL]	✓	✓
	LOL	6cdee600b5085c0c1d27c2a4d1654869	14	202	1	12	12	4	[GC][TS][U][NPP]	✓	✓
	Sausage Man	38fa5a9ba3a271ec9e2ad0724eae24d9	4	68	4	5	5	2	[IL][U]	✓	X
		217ac1c9109a9e0103d364a4356dbd40	14	527	10	15	70	2	[IL][U]	✓	✓
	PUBG2	bf111d5d095f9dc0d597cf0c93af7791	1	13	1	1	1	1	[UE4]	✓	X
		214d2b41ba49a3773c66befc0e1a4e4c	1	12	1	1	1	1	[GC]	✓	X
	SA Unity	-	1	8	8	1	8	1	[IL]	✓	✓
Windows	Assault Cube	5c0d8bfbb3589032f846cebb699993e1	1	23	1	7	7	1	[S]	✓	✓
	Bard's Tale	5dc6952102781bc2d8970d62f5d22a01	1	18	1	10	10	1	[S]	✓	✓
	Super Tux	42b9cafa7a6153d00fe2654ee01387e0	1	15	1	6	6	1	[S]	X	✓
	COD MW3	3a2d4279b71d30b9d29887a44335375b	1	411	1	9	9	1	[S]	X	✓

Detailed Result

Platform	Game	Hash	Input #Bin	Input ∑Size	Output Height	Output #Branch	Output #Edges	Output #Base	Output Bases	Memory Access Read	Memory Access Write
Android	CFM	4e349c25d1c5e303f73b9fa8b94934dd	1	652	1	3	3	2	[TS][U]	✓	X
	Arena 5v5	0a45409737c036f9d59c5feb427ba9c5	1	12	1	1	1	1	[GC]	✓	X
		2e8268d32dc22c31dd8579bca6b7f7d7	4	48	1	1	1	2	[GC][U]	✓	X
		* (10)	2	24	6	8	25	1	[GC]	✓	X
		...									
		c632aeaaecbe67487f0bf6f69416cb38	1	21	8	16	55	2	[GC][U]	✓	X
		76bab2ee423c05c1b6f10abc52653683	1	358	10	14	122	2	[GC][IL]	✓	✓
		04caab0a8f0b10d7750ae1d424034a7b	3	41	10	22	74	3	[GC][U][IL]	✓	X
	PUBG	397446459fe284a2c10f676b57c03982	1	13	1	1	1	1	[UE4]	✓	✓
		...									
		9a2cec9ac23cc6b9713d983d202a04ed	1	13	5	1	5	1	[UE4]	✓	✓
	COD	* (3)	1	12	1	1	1	1	[IL]	✓	X
		ac97f45290f238e5346f0ef5ae839cb9	21	269	1	3	3	3	[IL][GC][U]	✓	✓
	Royal Match	c8b4767dc7a0b57ce608173cbc7e6b15	1	8	6	1	7	1	[IL]	✓	✓
	LOL	6cdee600b5085c0c1d27c2a4d1654869	14	202	1	12	12	4	[GC][TS][U][NPP]	✓	✓
	Sausage Man	38fa5a9ba3a271ec9e2ad0724eae24d9	4	68	4	5	5	2	[IL][U]	✓	X
		217ac1c9109a9e0103d364a4356dbd40	14	527	10	15	70	2	[IL][U]	✓	✓
	PUBG2	bf111d5d095f9dc0d597cf0c93af7791	1	13	1	1	1	1	[UE4]	✓	X
		214d2b41ba49a3773c66befc0e1a4e4c	1	12	1	1	1	1	[GC]	✓	X
	SA Unity	-	1	8	8	1	8	1	[IL]	✓	✓
Windows	Assault Cube	5c0d8bfbb3589032f846cebb699993e1	1	23	1	7	7	1	[S]	✓	✓
	Bard's Tale	5dc6952102781bc2d8970d62f5d22a01	1	18	1	10	10	1	[S]	✓	✓
	Super Tux	42b9cafa7a6153d00fe2654ee01387e0	1	15	1	6	6	1	[S]	X	✓
	COD MW3	3a2d4279b71d30b9d29887a44335375b	1	411	1	9	9	1	[S]	X	✓

Detailed Result

Platform	Game	Hash	Input #Bin	Input ∑Size	Output Height	Output #Branch	Output #Edges	Output #Base	Output Bases	Memory Access Read	Memory Access Write
Android	CFM	4e349c25d1c5e303f73b9fa8b94934dd	1	652	1	3	3	2	[TS][U]	✓	X
	Arena 5v5	0a45409737c036f9d59c5feb427ba9c5	1	12	1	1	1	1	[GC]	✓	X
		2e8268d32dc22c31dd8579bca6b7f7d7	4	48	1	1	1	2	[GC][U]	✓	X
		* (10)	2	24	6	8	25	1	[GC]	✓	X
		...									
		c632aeaaecbe67487f0bf6f69416cb38	1	21	8	16	55	2	[GC][U]	✓	X
		76bab2ee423c05c1b6f10abc52653683	1	358	10	14	122	2	[GC][IL]	✓	✓
		04caab0a8f0b10d7750ae1d424034a7b	3	41	10	22	74	3	[GC][U][IL]	✓	X
	PUBG	397446459fe284a2c10f676b57c03982	1	13	1	1	1	1	[UE4]	✓	✓
		...									
		9a2cec9ac23cc6b9713d983d202a04ed	1	13	5	1	5	1	[UE4]	✓	✓
	COD	* (3)	1	12	1	1	1	1	[IL]	✓	X
		ac97f45290f238e5346f0ef5ae839cb9	21	269	1	3	3	3	[IL][GC][U]	✓	✓
	Royal Match	c8b4767dc7a0b57ce608173cbc7e6b15	1	8	6	1	7	1	[IL]	✓	✓
	LOL	6cdee600b5085c0c1d27c2a4d1654869	14	202	1	12	12	4	[GC][TS][U][NPP]	✓	✓
	Sausage Man	38fa5a9ba3a271ec9e2ad0724eae24d9	4	68	4	5	5	2	[IL][U]	✓	X
		217ac1c9109a9e0103d364a4356dbd40	14	527	10	15	70	2	[IL][U]	✓	✓
	PUBG2	bf111d5d095f9dc0d597cf0c93af7791	1	13	1	1	1	1	[UE4]	✓	X
		214d2b41ba49a3773c66befc0e1a4e4c	1	12	1	1	1	1	[GC]	✓	X
	SA Unity	-	1	8	8	1	8	1	[IL]	✓	✓
Windows	Assault Cube	5c0d8bfbb3589032f846cebb699993e1	1	23	1	7	7	1	[S]	✓	✓
	Bard's Tale	5dc6952102781bc2d8970d62f5d22a01	1	18	1	10	10	1	[S]	✓	✓
	Super Tux	42b9cafa7a6153d00fe2654ee01387e0	1	15	1	6	6	1	[S]	X	✓
	COD MW3	3a2d4279b71d30b9d29887a44335375b	1	411	1	9	9	1	[S]	X	✓

Detailed Result

Platform	Game	Hash	Input #Bin	Input ∑Size	Output Height	Output #Branch	Output #Edges	Output #Base	Output Bases	Memory Access Read	Memory Access Write
Android	CFM	4e349c25d1c5e303f73b9fa8b94934dd	1	652	1	3	3	2	[TS][U]	✓	✗
	Arena 5v5	0a45409737c036f9d59c5feb427ba9c5	1	12	1	1	1	1	[GC]	✓	✗
		2e8268d32dc22c31dd8579bca6b7f7d7	4	48	1	1	1	2	[GC][U]	✓	✗
		* (10)	2	24	6	8	25	1	[GC]	✓	✗
		...									
		c632aeaaecbe67487f0bf6f69416cb38	1	21	8	16	55	2	[GC][U]	✓	✗
		76bab2ee423c05c1b6f10abc52653683	1	358	10	14	122	2	[GC][IL]	✓	✓
		04caab0a8f0b10d7750ae1d424034a7b	3	41	10	22	74	3	[GC][U][IL]	✓	✗
	PUBG	397446459fe284a2c10f676b57c03982	1	13	1	1	1	1	[UE4]	✓	✓
		...									
		9a2cec9ac23cc6b9713d983d202a04ed	1	13	5	1	5	1	[UE4]	✓	✓
	COD	* (3)	1	12	1	1	1	1	[IL]	✓	✗
		ac97f45290f238e5346f0ef5ae839cb9	21	269	1	3	3	3	[IL][GC][U]	✓	✓
	Royal Match	c8b4767dc7a0b57ce608173cbc7e6b15	1	8	6	1	7	1	[IL]	✓	✓
	LOL	6cdee600b5085c0c1d27c2a4d1654869	14	202	1	12	12	4	[GC][TS][U][NPP]	✓	✓
	Sausage Man	38fa5a9ba3a271ec9e2ad0724eae24d9	4	68	4	5	5	2	[IL][U]	✓	✗
		217ac1c9109a9e0103d364a4356dbd40	14	527	10	15	70	2	[IL][U]	✓	✓
	PUBG2	bf111d5d095f9dc0d597cf0c93af7791	1	13	1	1	1	1	[UE4]	✓	✗
		214d2b41ba49a3773c66befc0e1a4e4c	1	12	1	1	1	1	[GC]	✓	✗
	SA Unity	-	1	8	8	1	8	1	[IL]	✓	✓
Windows	Assault Cube	5c0d8bfbb3589032f846cebb699993e1	1	23	1	7	7	1	[S]	✓	✓
	Bard's Tale	5dc6952102781bc2d8970d62f5d22a01	1	18	1	10	10	1	[S]	✓	✓
	Super Tux	42b9cafa7a6153d00fe2654ee01387e0	1	15	1	6	6	1	[S]	✗	✓
	COD MW3	3a2d4279b71d30b9d29887a44335375b	1	411	1	9	9	1	[S]	✗	✓

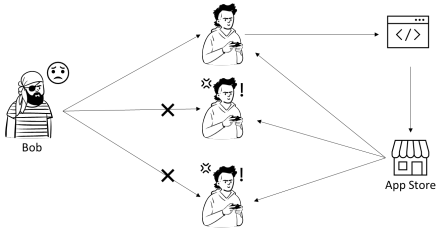
Detailed Result

Platform	Game	Hash	Input #Bin	Input ∑Size	Output Height	Output #Branch	Output #Edges	Output #Base	Output Bases	Memory Access Read	Memory Access Write
Android	CFM	4e349c25d1c5e303f73b9fa8b94934dd	1	652	1	3	3	2	[TS][U]	✓	✗
	Arena 5v5	0a45409737c036f9d59c5feb427ba9c5	1	12	1	1	1	1	[GC]	✓	✗
		2e8268d32dc22c31dd8579bca6b7f7d7	4	48	1	1	1	2	[GC][U]	✓	✗
		* (10)	2	24	6	8	25	1	[GC]	✓	✗
		...									
		c632aeaaecbe67487f0bf6f69416cb38	1	21	8	16	55	2	[GC][U]	✓	✗
		76bab2ee423c05c1b6f10abc52653683	1	358	10	14	122	2	[GC][IL]	✓	✓
		04caab0a8f0b10d7750ae1d424034a7b	3	41	10	22	74	3	[GC][U][IL]	✓	✗
	PUBG	397446459fe284a2c10f676b57c03982	1	13	1	1	1	1	[UE4]	✓	✓
		...									
		9a2cec9ac23cc6b9713d983d202a04ed	1	13	5	1	5	1	[UE4]	✓	✓
	COD	* (3)	1	12	1	1	1	1	[IL]	✓	✗
		ac97f45290f238e5346f0ef5ae839cb9	21	269	1	3	3	3	[IL][GC][U]	✓	✓
	Royal Match	c8b4767dc7a0b57ce608173cbc7e6b15	1	8	6	1	7	1	[IL]	✓	✓
	LOL	6cdee600b5085c0c1d27c2a4d1654869	14	202	1	12	12	4	[GC][TS][U][NPP]	✓	✓
	Sausage Man	38fa5a9ba3a271ec9e2ad0724eae24d9	4	68	4	5	5	2	[IL][U]	✓	✗
		217ac1c9109a9e0103d364a4356dbd40	14	527	10	15	70	2	[IL][U]	✓	✓
	PUBG2	bf111d5d095f9dc0d597cf0c93af7791	1	13	1	1	1	1	[UE4]	✓	✗
		214d2b41ba49a3773c66befc0e1a4e4c	1	12	1	1	1	1	[GC]	✓	✗
	SA Unity	-	1	8	8	1	8	1	[IL]	✓	✓
Windows	Assault Cube	5c0d8bfbb3589032f846cebb699993e1	1	23	1	7	7	1	[S]	✓	✓
	Bard's Tale	5dc6952102781bc2d8970d62f5d22a01	1	18	1	10	10	1	[S]	✓	✓
	Super Tux	42b9cafa7a6153d00fe2654ee01387e0	1	15	1	6	6	1	[S]	✗	✓
	COD MW3	3a2d4279b71d30b9d29887a44335375b	1	411	1	9	9	1	[S]	✗	✓

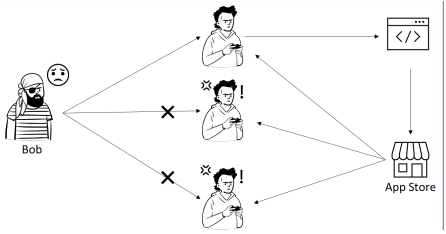
Detailed Result

Platform	Game	Hash	Input #Bin	Input ∑Size	Output Height	Output #Branch	Output #Edges	Output #Base	Output Bases	Memory Access Read	Memory Access Write
Android	CFM	4e349c25d1c5e303f73b9fa8b94934dd	1	652	1	3	3	2	[TS][U]	✓	X
	Arena 5v5	0a45409737c036f9d59c5feb427ba9c5	1	12	1	1	1	1	[GC]	✓	X
		2e8268d32dc22c31dd8579bca6b7f7d7	4	48	1	1	1	2	[GC][U]	✓	X
		* (10)	2	24	6	8	25	1	[GC]	✓	X
		...									
		c632aeaaecbe67487f0bf6f69416cb38	1	21	8	16	55	2	[GC][U]	✓	X
		76bab2ee423c05c1b6f10abc52653683	1	358	10	14	122	2	[GC][IL]	✓	✓
		04caab0a8f0b10d7750ae1d424034a7b	3	41	10	22	74	3	[GC][U][IL]	✓	X
	PUBG	397446459fe284a2c10f676b57c03982	1	13	1	1	1	1	[UE4]	✓	✓
		...									
		9a2cec9ac23cc6b9713d983d202a04ed	1	13	5	1	5	1	[UE4]	✓	✓
	COD	* (3)	1	12	1	1	1	1	[IL]	✓	X
		ac97f45290f238e5346f0ef5ae839cb9	21	269	1	3	3	3	[IL][GC][U]	✓	✓
	Royal Match	c8b4767dc7a0b57ce608173cbc7e6b15	1	8	6	1	7	1	[IL]	✓	✓
	LOL	6cdee600b5085c0c1d27c2a4d1654869	14	202	1	12	12	4	[GC][TS][U][NPP]	✓	✓
	Sausage Man	38fa5a9ba3a271ec9e2ad0724eae24d9	4	68	4	5	5	2	[IL][U]	✓	X
		217ac1c9109a9e0103d364a4356dbd40	14	527	10	15	70	2	[IL][U]	✓	✓
	PUBG2	bf111d5d095f9dc0d597cf0c93af7791	1	13	1	1	1	1	[UE4]	✓	X
		214d2b41ba49a3773c66befc0e1a4e4c	1	12	1	1	1	1	[GC]	✓	X
	SA Unity	-	1	8	8	1	8	1	[IL]	✓	✓
Windows	Assault Cube	5c0d8bfbb3589032f846cebb699993e1	1	23	1	7	7	1	[S]	✓	✓
	Bard's Tale	5dc6952102781bc2d8970d62f5d22a01	1	18	1	10	10	1	[S]	✓	✓
	Super Tux	42b9cafa7a6153d00fe2654ee01387e0	1	15	1	6	6	1	[S]	X	✓
	COD MW3	3a2d4279b71d30b9d29887a44335375b	1	411	1	9	9	1	[S]	X	✓

Takeaway

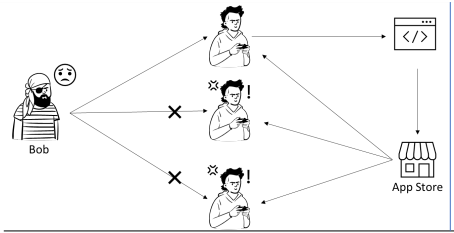


Takeaway

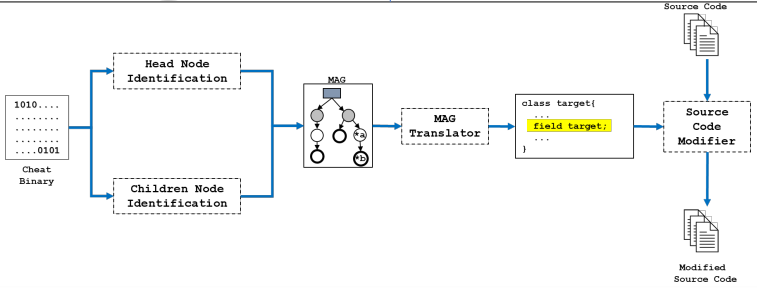


Victim Game	Platform	Release	#Installs	Engine	#Cheats	#Binary
CrossFire Mobile	Android	12/3/2015	-	Unity	1	1
Arena 5v5	Android	11/30/2016	10M+	Unity	29	55
PUBG Mobile	Android	3/23/2017	500M+	UE4	35	74
COD Mobile	Android	10/1/2019	100M+	Unity	4	24
Royal Match	Android	2/25/2021	10M+	Unity	1	1
LOL	Android	10/27/2021	-	Unity	1	14
PUBG New State	Android	11/11/2021	10M+	UE4	2	2
Sausage Man	Android	4/29/2022	10M+	Unity	2	18
Assault Cube	Windows	4/1/2022	-	CUBE	1	1
Bard's Tale	Windows	6/17/2005	-	Dark Alliance	1	1
Super Tux	Windows	12/22/2021	-	SuperTux	1	1
COD MW3	Windows	11/8/2011	-	IW	1	1

Takeaway



Victim Game	Platform	Release	#Installs	Engine	#Cheats	#Binary
CrossFire Mobile	Android	12/3/2015	-	Unity	1	1
Arena 5v5	Android	11/30/2016	10M+	Unity	29	55
PUBG Mobile	Android	3/23/2017	500M+	UE4	35	74
COD Mobile	Android	10/1/2019	100M+	Unity	4	24
Royal Match	Android	2/25/2021	10M+	Unity	1	1
LOL	Android	10/27/2021	-	Unity	1	14
PUBG New State	Android	11/11/2021	10M+	UE4	2	2
Sausage Man	Android	4/29/2022	10M+	Unity	2	18
Assault Cube	Windows	4/1/2022	-	CUBE	1	1
Bard's Tale	Windows	6/17/2005	-	Dark Alliance	1	1
Super Tux	Windows	12/22/2021	-	SuperTux	1	1
COD MW3	Windows	11/8/2011	-	IW	1	1



Q&A

CheatFighter Source Code
https://github.com/OSUSecLab/CheatFighter

SecLab @ OSU
https://go.osu.edu/seclab